

```
<!DOCTYPE html>
<html lang="en" >
<head>
  <title>Latihan Koding Informatika MAN Kapuas</title>

<style>
p {
  font-size: 13px;
}

.headingcaption {
  color: #202569;
  text-align: center;
  font-size: 2.3em;
  text-transform: uppercase;
  padding: 40px 20px;
}

.boxtitle {
  width: 100%;
  color: #202569;
  margin: 0;
  text-align: center;
  position: absolute;
  top: -20px;
  z-index: 1;
  font-size: 2em;
}

.boxtitle span {
  background-color: #fff;
  padding-left: 10px;
  padding-right: 10px;
}

.savonatrim {
  background-color: #a10000;
  padding-left: 3px;
  padding-right: 3px;
  padding-bottom: 3px;
  margin-top: 20px;
}

.savonabox {
  background-color: #fff;
  color: #000;
  margin-top: 3px;
```

```
padding: 20px;  
font-size: 1.2em;  
}
```

```
.wyndhamtrim {  
background-color: #a10000;  
padding-left: 3px;  
padding-right: 3px;  
padding-bottom: 3px;  
margin-top: 20px;  
}
```

```
.wyndhambox {  
background-color: #fff;  
color: #000;  
margin-top: 3px;  
padding: 20px;  
font-size: 1.2em;  
}
```

```
.logocenter {  
display: block;  
margin-left: auto;  
margin-right: auto;  
width: 50%;  
padding: 20px 0;  
}
```

```
.limitfooter {  
font-size: 1.5em;  
color: #000;  
text-align: center;  
padding: 30px 0;  
}
```

```
#quantity {  
font-size: 1.2em;  
color: #202569;  
font-weight: 600;  
text-align: center;  
padding: 10px 0;  
}
```

```
#storenearyou {  
font-size: 1.1em;  
color: #202569;  
font-weight: 600;  
text-align: center;
```

```
padding: 10px 0 20px 0;
text-transform: uppercase;
}
```

```
.flip-clock-wrapper * {
  -webkit-box-sizing: border-box;
  -moz-box-sizing: border-box;
  -ms-box-sizing: border-box;
  -o-box-sizing: border-box;
  box-sizing: border-box;
  -webkit-backface-visibility: hidden;
  -moz-backface-visibility: hidden;
  -ms-backface-visibility: hidden;
  -o-backface-visibility: hidden;
  backface-visibility: hidden;
}
```

```
.flip-clock-wrapper a {
  cursor: pointer;
  text-decoration: none;
  color: #ccc;
}
```

```
.flip-clock-wrapper a:hover {
  color: #fff;
}
```

```
.flip-clock-wrapper ul {
  list-style: none;
}
```

```
.flip-clock-wrapper.clearfix:before,
.flip-clock-wrapper.clearfix:after {
  content: " ";
  display: table;
}
```

```
.flip-clock-wrapper.clearfix:after {
  clear: both;
}
```

```
.flip-clock-wrapper.clearfix {
  *zoom: 1;
}
```

```
/* Main */
.flip-clock-wrapper {
  font: normal 10px "Helvetica Neue", Helvetica, sans-serif;
```

```
-webkit-user-select: none;  
}
```

```
.flip-clock-meridium {  
  background: none !important;  
  box-shadow: 0 0 0 !important;  
  font-size: 36px !important;  
}
```

```
.flip-clock-meridium a {  
  color: #313333;  
}
```

```
.flip-clock-wrapper {  
  text-align: center;  
  position: relative;  
  width: 100%;  
  margin: 1em;  
}
```

```
.flip-clock-wrapper:before,  
.flip-clock-wrapper:after {  
  content: " "; /* 1 */  
  display: table; /* 2 */  
}  
.flip-clock-wrapper:after {  
  clear: both;  
}
```

```
/* Skeleton */
```

```
.flip-clock-wrapper ul {  
  position: relative;  
  float: left;  
  margin: 5px;  
  width: 60px;  
  height: 90px;  
  font-size: 80px;  
  font-weight: bold;  
  line-height: 87px;  
  border-radius: 6px;  
  background: #000;  
}
```

```
.flip-clock-wrapper ul li {  
  z-index: 1;  
  position: absolute;  
  left: 0;  
  top: 0;
```

```

width: 100%;
height: 100%;
line-height: 87px;
text-decoration: none !important;
}

.flip-clock-wrapper ul li:first-child {
  z-index: 2;
}

.flip-clock-wrapper ul li a {
  display: block;
  height: 100%;
  -webkit-perspective: 200px;
  -moz-perspective: 200px;
  perspective: 200px;
  margin: 0 !important;
  overflow: visible !important;
  cursor: default !important;
}

.flip-clock-wrapper ul li a div {
  z-index: 1;
  position: absolute;
  left: 0;
  width: 100%;
  height: 50%;
  font-size: 80px;
  overflow: hidden;
  outline: 1px solid transparent;
}

.flip-clock-wrapper ul li a div .shadow {
  position: absolute;
  width: 100%;
  height: 100%;
  z-index: 2;
}

.flip-clock-wrapper ul li a div.up {
  -webkit-transform-origin: 50% 100%;
  -moz-transform-origin: 50% 100%;
  -ms-transform-origin: 50% 100%;
  -o-transform-origin: 50% 100%;
  transform-origin: 50% 100%;
  top: 0;
}

```

```
.flip-clock-wrapper ul li a div.up:after {  
  content: "";  
  position: absolute;  
  top: 44px;  
  left: 0;  
  z-index: 5;  
  width: 100%;  
  height: 3px;  
  background-color: #000;  
  background-color: rgba(0, 0, 0, 0.4);  
}
```

```
.flip-clock-wrapper ul li a div.down {  
  -webkit-transform-origin: 50% 0;  
  -moz-transform-origin: 50% 0;  
  -ms-transform-origin: 50% 0;  
  -o-transform-origin: 50% 0;  
  transform-origin: 50% 0;  
  bottom: 0;  
  border-bottom-left-radius: 6px;  
  border-bottom-right-radius: 6px;  
}
```

```
.flip-clock-wrapper ul li a div div.inn {  
  position: absolute;  
  left: 0;  
  z-index: 1;  
  width: 100%;  
  height: 200%;  
  color: #ccc;  
  text-shadow: 0 1px 2px #000;  
  text-align: center;  
  background-color: #333;  
  border-radius: 6px;  
  font-size: 70px;  
}
```

```
.flip-clock-wrapper ul li a div.up div.inn {  
  top: 0;  
}
```

```
.flip-clock-wrapper ul li a div.down div.inn {  
  bottom: 0;  
}
```

```
/* PLAY */
```

```
.flip-clock-wrapper ul.play li.flip-clock-before {  
  z-index: 3;
```

```
}
```

```
.flip-clock-wrapper .flip {  
  box-shadow: 0 2px 5px rgba(0, 0, 0, 0.7);  
}
```

```
.flip-clock-wrapper ul.play li.flip-clock-active {  
  -webkit-animation: asd 0.01s 0.49s linear both;  
  -moz-animation: asd 0.01s 0.49s linear both;  
  animation: asd 0.01s 0.49s linear both;  
  z-index: 5;  
}
```

```
.flip-clock-divider {  
  float: left;  
  display: inline-block;  
  position: relative;  
  width: 20px;  
  height: 100px;  
}
```

```
.flip-clock-divider:first-child {  
  width: 0;  
}
```

```
.flip-clock-dot {  
  display: block;  
  background: #323434;  
  width: 10px;  
  height: 10px;  
  position: absolute;  
  border-radius: 50%;  
  box-shadow: 0 0 5px rgba(0, 0, 0, 0.5);  
  left: 5px;  
}
```

```
.flip-clock-divider .flip-clock-label {  
  position: absolute;  
  top: -1.5em;  
  right: -85px;  
  color: black;  
  text-shadow: none;  
}
```

```
.flip-clock-divider.hours .flip-clock-label {  
  right: -89px;  
}
```

```
.flip-clock-divider.minutes .flip-clock-label {  
  right: -93px;  
}
```

```
.flip-clock-divider.seconds .flip-clock-label {  
  right: -94px;  
}
```

```
.flip-clock-dot.top {  
  top: 30px;  
}
```

```
.flip-clock-dot.bottom {  
  bottom: 30px;  
}
```

```
@-webkit-keyframes asd {  
  0% {  
    z-index: 2;  
  }  
  
  100% {  
    z-index: 4;  
  }  
}
```

```
@-moz-keyframes asd {  
  0% {  
    z-index: 2;  
  }  
  
  100% {  
    z-index: 4;  
  }  
}
```

```
@-o-keyframes asd {  
  0% {  
    z-index: 2;  
  }  
  
  100% {  
    z-index: 4;  
  }  
}
```

```
@keyframes asd {  
  0% {
```



```
    z-index: 2;
}

100% {
    z-index: 4;
}
}

.flip-clock-wrapper ul.play li.flip-clock-active .down {
    z-index: 2;
    -webkit-animation: turn 0.5s 0.5s linear both;
    -moz-animation: turn 0.5s 0.5s linear both;
    animation: turn 0.5s 0.5s linear both;
}

@-webkit-keyframes turn {
    0% {
        -webkit-transform: rotateX(90deg);
    }

    100% {
        -webkit-transform: rotateX(0deg);
    }
}

@-moz-keyframes turn {
    0% {
        -moz-transform: rotateX(90deg);
    }

    100% {
        -moz-transform: rotateX(0deg);
    }
}

@-o-keyframes turn {
    0% {
        -o-transform: rotateX(90deg);
    }

    100% {
        -o-transform: rotateX(0deg);
    }
}

@keyframes turn {
    0% {
        transform: rotateX(90deg);
```

```

}

100% {
  transform: rotateX(0deg);
}
}

.flip-clock-wrapper ul.play li.flip-clock-before .up {
  z-index: 2;
  -webkit-animation: turn2 0.5s linear both;
  -moz-animation: turn2 0.5s linear both;
  animation: turn2 0.5s linear both;
}

@-webkit-keyframes turn2 {
  0% {
    -webkit-transform: rotateX(0deg);
  }

  100% {
    -webkit-transform: rotateX(-90deg);
  }
}

@-moz-keyframes turn2 {
  0% {
    -moz-transform: rotateX(0deg);
  }

  100% {
    -moz-transform: rotateX(-90deg);
  }
}

@-o-keyframes turn2 {
  0% {
    -o-transform: rotateX(0deg);
  }

  100% {
    -o-transform: rotateX(-90deg);
  }
}

@keyframes turn2 {
  0% {
    transform: rotateX(0deg);
  }
}

```

```

100% {
    transform: rotateX(-90deg);
}
}

.flip-clock-wrapper ul li.flip-clock-active {
    z-index: 3;
}

/* SHADOW */
.flip-clock-wrapper ul.play li.flip-clock-before .up .shadow {
    background: -moz-linear-gradient(top, rgba(0, 0, 0, 0.1) 0%, black 100%);
    background: -webkit-gradient(
        linear,
        left top,
        left bottom,
        color-stop(0%, rgba(0, 0, 0, 0.1)),
        color-stop(100%, black)
    );
    background: linear, top, rgba(0, 0, 0, 0.1) 0%, black 100%;
    background: -o-linear-gradient(top, rgba(0, 0, 0, 0.1) 0%, black 100%);
    background: -ms-linear-gradient(top, rgba(0, 0, 0, 0.1) 0%, black 100%);
    background: linear, to bottom, rgba(0, 0, 0, 0.1) 0%, black 100%;
    -webkit-animation: show 0.5s linear both;
    -moz-animation: show 0.5s linear both;
    animation: show 0.5s linear both;
}

.flip-clock-wrapper ul.play li.flip-clock-active .up .shadow {
    background: -moz-linear-gradient(top, rgba(0, 0, 0, 0.1) 0%, black 100%);
    background: -webkit-gradient(
        linear,
        left top,
        left bottom,
        color-stop(0%, rgba(0, 0, 0, 0.1)),
        color-stop(100%, black)
    );
    background: linear, top, rgba(0, 0, 0, 0.1) 0%, black 100%;
    background: -o-linear-gradient(top, rgba(0, 0, 0, 0.1) 0%, black 100%);
    background: -ms-linear-gradient(top, rgba(0, 0, 0, 0.1) 0%, black 100%);
    background: linear, to bottom, rgba(0, 0, 0, 0.1) 0%, black 100%;
    -webkit-animation: hide 0.5s 0.3s linear both;
    -moz-animation: hide 0.5s 0.3s linear both;
    animation: hide 0.5s 0.3s linear both;
}

/*DOWN*/

```

```
.flip-clock-wrapper ul.play li.flip-clock-before .down .shadow {
  background: -moz-linear-gradient(top, black 0%, rgba(0, 0, 0, 0.1) 100%);
  background: -webkit-gradient(
    linear,
    left top,
    left bottom,
    color-stop(0%, black),
    color-stop(100%, rgba(0, 0, 0, 0.1))
  );
  background: linear, top, black 0%, rgba(0, 0, 0, 0.1) 100%;
  background: -o-linear-gradient(top, black 0%, rgba(0, 0, 0, 0.1) 100%);
  background: -ms-linear-gradient(top, black 0%, rgba(0, 0, 0, 0.1) 100%);
  background: linear, to bottom, black 0%, rgba(0, 0, 0, 0.1) 100%;
  -webkit-animation: show 0.5s linear both;
  -moz-animation: show 0.5s linear both;
  animation: show 0.5s linear both;
}
```

```
.flip-clock-wrapper ul.play li.flip-clock-active .down .shadow {
  background: -moz-linear-gradient(top, black 0%, rgba(0, 0, 0, 0.1) 100%);
  background: -webkit-gradient(
    linear,
    left top,
    left bottom,
    color-stop(0%, black),
    color-stop(100%, rgba(0, 0, 0, 0.1))
  );
  background: linear, top, black 0%, rgba(0, 0, 0, 0.1) 100%;
  background: -o-linear-gradient(top, black 0%, rgba(0, 0, 0, 0.1) 100%);
  background: -ms-linear-gradient(top, black 0%, rgba(0, 0, 0, 0.1) 100%);
  background: linear, to bottom, black 0%, rgba(0, 0, 0, 0.1) 100%;
  -webkit-animation: hide 0.5s 0.3s linear both;
  -moz-animation: hide 0.5s 0.3s linear both;
  animation: hide 0.5s 0.2s linear both;
}
```

```
@-webkit-keyframes show {
  0% {
    opacity: 0;
  }

  100% {
    opacity: 1;
  }
}
```

```
@-moz-keyframes show {
  0% {
```

```
    opacity: 0;
}

100% {
    opacity: 1;
}
}

@-o-keyframes show {
    0% {
        opacity: 0;
    }

    100% {
        opacity: 1;
    }
}

@keyframes show {
    0% {
        opacity: 0;
    }

    100% {
        opacity: 1;
    }
}

@-webkit-keyframes hide {
    0% {
        opacity: 1;
    }

    100% {
        opacity: 0;
    }
}

@-moz-keyframes hide {
    0% {
        opacity: 1;
    }

    100% {
        opacity: 0;
    }
}
```

```
@-o-keyframes hide {
  0% {
    opacity: 1;
  }

  100% {
    opacity: 0;
  }
}
```

```
@keyframes hide {
  0% {
    opacity: 1;
  }

  100% {
    opacity: 0;
  }
}
```

```
@import url("https://fonts.googleapis.com/css?family=Poppins:400,400i,700,700i");
.count-up-section {
  color: #222;
  font-family: "Poppins", sans-serif;
  font-size: 16px;
  line-height: 1.48;
  padding: 5vh 0 0 0;
}
.copyright.text-center {
  text-align: center;
}
.count-up-section .style-point {
  color: #000;
  font-size: 38px;
}
.count-up-section .style-point span {
  background-color: #3e4b99;
  background-image: -moz-linear-gradient(90deg, #3e4b99 10%, #d81e1e 100%);
  background-image: -webkit-linear-gradient(90deg, #3e4b99 10%, #d81e1e 100%);
  background-image: -o-linear-gradient(90deg, #3e4b99 10%, #d81e1e 100%);
  background-image: -ms-linear-gradient(90deg, #3e4b99 10%, #d81e1e 100%);
  background-image: linear-gradient(90deg, #3e4b99 10%, #d81e1e 100%);
  color: #fff;
  border-radius: 50%;
  display: block;
  width: 80px;
  height: 80px;
  text-align: center;
```

```

    font-size: 54px;
}
.count-up-section .colorfont {
    color: #0147fe;
}
#count-up-section6.count-up-section {
    background-color: #000;
    max-width: 100%;
}
.flipTimemodulesboxes {
    max-width: 720px;
    margin: 0 auto;
    padding: 0 15px;
}
.darkbgFlip h2.style-point {
    color: #fff;
}
.flipTimebox {
    margin: 50px auto 100px;
}

/*days, hours, minutes, seconds styling*/
.flipTimebox .flip-clock-divider .flip-clock-label {
    font-size: 1em;
    font-weight: bold;
    text-transform: uppercase;
    font-family: arial;
}
.flipTimebox .flip-clock-wrapper ul li a div.up:after {
    background-color: rgba(128, 128, 128, 0.5);
}
.flipTimebox .flip-clock-wrapper .flip-clock-meridium a {
    color: #fff;
    background-color: #333333;
    border-radius: 6px;
}
.flipclock1message,
.flipclock2message,
.flipclock3message,
.flipclock4message,
.flipclock5message,
.flipclock6message,
.flipclock14message {
    padding-left: 15px;
    color: #ff0000;
}
.darkbgFlip .flip-clock-divider .flip-clock-label {
    color: #fff;

```

```

}
.flipTimebox .orangetheme.flip-clock-wrapper ul li a div div.inn {
  color: #fff;
  background-color: #fb9902;
}
.flipTimebox .orangetheme .flip-clock-dot {
  background-color: #fb9902;
  box-shadow: none;
}
.flipTimebox .orangetheme.flip-clock-wrapper .flip-clock-meridium a {
  color: #fff;
  background-color: #fb9902;
  border-radius: 6px;
}
.flipTimebox .orangetheme.flip-clock-wrapper ul li a div.up:after {
  background-color: rgba(250, 250, 250, 0.5);
}
.flipTimebox .darkorangetheme.flip-clock-wrapper ul li a div div.inn {
  color: #fff;
  background-color: #fd5308;
}
.flipTimebox .darkorangetheme .flip-clock-dot {
  background-color: #fd5308;
  box-shadow: none;
}
.flipTimebox .darkorangetheme.flip-clock-wrapper .flip-clock-meridium a {
  color: #fff;
  background-color: #fd5308;
  border-radius: 6px;
}
.flipTimebox .darkorangetheme.flip-clock-wrapper ul li a div.up:after {
  background-color: rgba(250, 250, 250, 0.5);
}
.flipTimebox .bluetheme.flip-clock-wrapper ul li a div div.inn {
  color: #fff;
  background-color: #0147fe;
}
.flipTimebox .bluetheme .flip-clock-dot {
  background-color: #0147fe;
  box-shadow: none;
}
.flipTimebox .bluetheme.flip-clock-wrapper .flip-clock-meridium a {
  color: #fff;
  background-color: #0147fe;
  border-radius: 6px;
}
.flipTimebox .bluetheme.flip-clock-wrapper ul li a div.up:after {
  background-color: rgba(250, 250, 250, 0.5);
}

```



```

}
.flipTimebox .greentheme.flip-clock-wrapper ul li a div div.inn {
  color: #fff;
  background-color: #23a929;
}
.flipTimebox .greentheme .flip-clock-dot {
  background-color: #23a929;
  box-shadow: none;
}
.flipTimebox .greentheme.flip-clock-wrapper .flip-clock-meridium a {
  color: #fff;
  background-color: #23a929;
  border-radius: 6px;
}
.flipTimebox .greentheme.flip-clock-wrapper ul li a div.up:after {
  background-color: rgba(250, 250, 250, 0.5);
}
.flipTimebox .redtheme.flip-clock-wrapper ul li a div div.inn {
  color: #fff;
  background-color: #d81e1e;
}
.flipTimebox .redtheme .flip-clock-dot {
  background-color: #d81e1e;
  box-shadow: none;
}
.flipTimebox .redtheme.flip-clock-wrapper .flip-clock-meridium a {
  color: #fff;
  background-color: #d81e1e;
  border-radius: 6px;
}
.flipTimebox .redtheme.flip-clock-wrapper ul li a div.up:after {
  background-color: rgba(250, 250, 250, 0.5);
}
.flipTimebox .pinktheme.flip-clock-wrapper ul li a div div.inn {
  color: #fff;
  background-color: #8501ad;
}
.flipTimebox .pinktheme .flip-clock-dot {
  background-color: #8501ad;
  box-shadow: none;
}
.flipTimebox .pinktheme.flip-clock-wrapper .flip-clock-meridium a {
  color: #fff;
  background-color: #8501ad;
  border-radius: 6px;
}
.flipTimebox .pinktheme.flip-clock-wrapper ul li a div.up:after {
  background-color: rgba(250, 250, 250, 0.5);
}

```

```

}
.flipTimebox .purpletheme.flip-clock-wrapper ul li a div div.inn {
  color: #fff;
  background-color: #a7194b;
}
.flipTimebox .purpletheme .flip-clock-dot {
  background-color: #a7194b;
  box-shadow: none;
}
.flipTimebox .purpletheme.flip-clock-wrapper .flip-clock-meridium a {
  color: #fff;
  background-color: #a7194b;
  border-radius: 6px;
}
.flipTimebox .purpletheme.flip-clock-wrapper ul li a div.up:after {
  background-color: rgba(250, 250, 250, 0.5);
}
.flipTimebox .coffeetheme.flip-clock-wrapper ul li a div div.inn {
  color: #fff;
  background-color: #7b5a26;
}
.flipTimebox .coffeetheme .flip-clock-dot {
  background-color: #7b5a26;
}
.flipTimebox .coffeetheme.flip-clock-wrapper .flip-clock-meridium a {
  color: #fff;
  background-color: #7b5a26;
  border-radius: 6px;
}
.flipTimebox .coffeetheme.flip-clock-wrapper ul li a div.up:after {
  background-color: rgba(250, 250, 250, 0.5);
}
.flipTimebox .gradienttheme.flip-clock-wrapper ul li a div div.inn {
  color: #fff;
  background-color: #3e4b99;
  background-image: -moz-linear-gradient(90deg, #3e4b99 10%, #d81e1e 100%);
  background-image: -webkit-linear-gradient(90deg, #3e4b99 10%, #d81e1e 100%);
  background-image: -o-linear-gradient(90deg, #3e4b99 10%, #d81e1e 100%);
  background-image: -ms-linear-gradient(90deg, #3e4b99 10%, #d81e1e 100%);
  background-image: linear-gradient(90deg, #3e4b99 10%, #d81e1e 100%);
}
.flipTimebox .gradienttheme .flip-clock-dot {
  box-shadow: none;
  background-color: #3e4b99;
  background-image: -moz-linear-gradient(90deg, #3e4b99 10%, #d81e1e 100%);
  background-image: -webkit-linear-gradient(90deg, #3e4b99 10%, #d81e1e 100%);
  background-image: -o-linear-gradient(90deg, #3e4b99 10%, #d81e1e 100%);
  background-image: -ms-linear-gradient(90deg, #3e4b99 10%, #d81e1e 100%);
}

```

```

    background-image: linear-gradient(90deg, #3e4b99 10%, #d81e1e 100%);
}
.flipTimebox .gradienttheme.flip-clock-wrapper .flip-clock-meridium a {
    color: #fff;
    border-radius: 6px;
    background-color: #3e4b99;
    background-image: -moz-linear-gradient(90deg, #3e4b99 10%, #d81e1e 100%);
    background-image: -webkit-linear-gradient(90deg, #3e4b99 10%, #d81e1e 100%);
    background-image: -o-linear-gradient(90deg, #3e4b99 10%, #d81e1e 100%);
    background-image: -ms-linear-gradient(90deg, #3e4b99 10%, #d81e1e 100%);
    background-image: linear-gradient(90deg, #3e4b99 10%, #d81e1e 100%);
}
.flipTimebox .gradienttheme.flip-clock-wrapper ul li a div.up:after {
    background-color: rgba(250, 250, 250, 0.5);
}
.flipTimebox .gradienttheme2.flip-clock-wrapper ul li a div div.inn {
    color: #fff;
    background-color: #0147fe;
    background-image: -moz-linear-gradient(270deg, #0147fe 0%, #23a929 100%);
    background-image: -webkit-linear-gradient(270deg, #0147fe 0%, #23a929 100%);
    background-image: -o-linear-gradient(270deg, #0147fe 0%, #23a929 100%);
    background-image: -ms-linear-gradient(270deg, #0147fe 0%, #23a929 100%);
    background-image: linear-gradient(270deg, #0147fe 0%, #23a929 100%);
}
.flipTimebox .gradienttheme2 .flip-clock-dot {
    box-shadow: none;
    background-color: #0147fe;
    background-image: -moz-linear-gradient(270deg, #0147fe 0%, #23a929 100%);
    background-image: -webkit-linear-gradient(270deg, #0147fe 0%, #23a929 100%);
    background-image: -o-linear-gradient(270deg, #0147fe 0%, #23a929 100%);
    background-image: -ms-linear-gradient(270deg, #0147fe 0%, #23a929 100%);
    background-image: linear-gradient(270deg, #0147fe 0%, #23a929 100%);
}
.flipTimebox .gradienttheme2.flip-clock-wrapper .flip-clock-meridium a {
    color: #fff;
    border-radius: 6px;
    background-color: #0147fe;
    background-image: -moz-linear-gradient(270deg, #0147fe 0%, #23a929 100%);
    background-image: -webkit-linear-gradient(270deg, #0147fe 0%, #23a929 100%);
    background-image: -o-linear-gradient(270deg, #0147fe 0%, #23a929 100%);
    background-image: -ms-linear-gradient(270deg, #0147fe 0%, #23a929 100%);
    background-image: linear-gradient(270deg, #0147fe 0%, #23a929 100%);
}
.flipTimebox .gradienttheme2.flip-clock-wrapper ul li a div.up:after {
    background-color: rgba(250, 250, 250, 0.5);
}
.flipTimebox .whitetheme.flip-clock-wrapper ul li a div div.inn {
    color: #0147fe;

```

```

    background-color: #fff;
}
.flipTimebox .whitetheme .flip-clock-dot {
    background-color: #fff;
    box-shadow: none;
}
.flipTimebox .whitetheme.flip-clock-wrapper .flip-clock-meridium a {
    color: #0147fe;
    background-color: #fff;
    border-radius: 6px;
}
.flipTimebox .whitetheme.flip-clock-wrapper ul li a div.up:after {
    background-color: rgba(128, 128, 128, 0.5);
}
@media screen and (max-width: 767px) {
    .flipTimebox .flip-clock-wrapper ul {
        height: 50px;
        line-height: 50px;
    }
    .flipTimebox .flip-clock-wrapper ul li {
        line-height: 50px;
    }
    .flipTimebox .flip-clock-wrapper ul li a div.up:after {
        top: 24px;
    }
    .flipTimebox .flip-clock-divider {
        height: 50px;
    }
    .flipTimebox .flip-clock-dot {
        height: 6px;
        width: 6px;
        left: 7px;
    }
    .flipTimebox .flip-clock-dot.top {
        top: 17px;
    }
    .flipTimebox .flip-clock-dot.bottom {
        bottom: 8px;
    }
    .flipTimebox .flip-clock-divider.days .flip-clock-label {
        right: -58px;
    }
    .flipTimebox .flip-clock-divider.hours .flip-clock-label {
        right: -62px;
    }
    .flipTimebox .flip-clock-divider.minutes .flip-clock-label {
        right: -68px;
    }
}

```

```

.flipTimebox .flip-clock-divider.seconds .flip-clock-label {
  right: -69px;
}
.flipTimebox .flip-clock-wrapper ul {
  width: 37px;
  margin: 2px;
}
.flipTimebox .flip-clock-wrapper ul li a div div.inn {
  font-size: 38px;
}
.flipTimebox .flip-clock-meridium {
  font-size: 20px !important;
}
}
@media screen and (max-width: 568px) {
  .flipTimebox .flip-clock-wrapper ul {
    margin: 2px;
  }
  .flipTimebox .flip-clock-divider {
    width: 5px;
  }
  .flipTimebox .flip-clock-dot {
    height: 5px;
    width: 5px;
    left: 0;
  }
  .flipTimebox .flip-clock-divider.days .flip-clock-label {
    right: -58px;
  }
  .flipTimebox .flip-clock-divider.hours .flip-clock-label {
    right: -62px;
  }
  .flipTimebox .flip-clock-divider.minutes .flip-clock-label {
    right: -68px;
  }
  .flipTimebox .flip-clock-divider.seconds .flip-clock-label {
    right: -69px;
  }
  .flipTimebox .flip-clock-meridium {
    font-size: 18px !important;
  }
}
@media screen and (max-width: 480px) {
  .flipTimemodulesboxes {
    padding: 0;
  }
  .flipTimebox .flip-clock-divider .flip-clock-label {
    font-size: 1em;
  }
}

```

```

}
.flipTimebox .flip-clock-divider.days .flip-clock-label {
  right: -33px;
}
.flipTimebox .flip-clock-divider.hours .flip-clock-label {
  right: -37px;
}
.flipTimebox .flip-clock-divider.minutes .flip-clock-label {
  right: -43px;
}
.flipTimebox .flip-clock-divider.seconds .flip-clock-label {
  right: -45px;
}
.flipTimebox .flip-clock-wrapper ul li a div div.inn {
  font-size: 28px;
}
.flipTimebox .flip-clock-meridium {
  font-size: 14px !important;
}
.flipTimebox .flip-clock-wrapper ul.flip {
  width: 27px;
  background-color: transparent;
  box-shadow: none;
  margin-left: -8px;
}
.flipTimebox .flip-clock-wrapper ul li {
  width: 30px;
}
.flipTimebox .flip-clock-dot {
  margin-left: -10px;
}
.flipTimebox {
  margin: 50px 0 100px 20px;
}
}
@media screen and (max-width: 360px) {
  .flipTimebox .flip-clock-divider .flip-clock-label {
    font-size: 1em;
    width: 3px;
  }
  .flipTimebox .flip-clock-divider.days .flip-clock-label {
    right: -1px;
  }
  .flipTimebox .flip-clock-divider.hours .flip-clock-label {
    right: 2px;
  }
  .flipTimebox .flip-clock-divider.minutes .flip-clock-label {
    right: 6px;
  }

```

```

}
.flipTimebox .flip-clock-divider.seconds .flip-clock-label {
  right: 9px;
}
.flipTimebox .flip-clock-wrapper ul.flip {
  width: 25px;
  margin-left: -14px;
}
.flipTimebox .flip-clock-wrapper {
  padding-left: 5px;
}
.flipTimebox .flip-clock-wrapper ul li {
  width: 25px;
}
.flipTimebox .flip-clock-dot {
  margin-left: -16px;
}
.flipTimebox {
  margin: 50px 0 100px 20px;
}
}
@media screen and (max-width: 320px) {
  .flipTimebox .flip-clock-wrapper {
    padding-left: 5px;
  }
  .flipTimebox .flip-clock-wrapper ul.flip {
    width: 22px;
    margin-left: -16px;
  }
  .flipTimebox .flip-clock-wrapper ul li {
    width: 22px;
  }
  .flipTimebox .flip-clock-dot {
    margin-left: -18px;
  }
  .flipTimebox .flip-clock-divider.days .flip-clock-label {
    right: 3px;
  }
  .flipTimebox .flip-clock-divider.hours .flip-clock-label {
    right: 6px;
  }
  .flipTimebox .flip-clock-divider.minutes .flip-clock-label {
    right: 11px;
  }
  .flipTimebox .flip-clock-divider.seconds .flip-clock-label {
    right: 14px;
  }
}
}

```

```

</style>
</head>
<body>
<!-- partial:index.partial.html -->
<body>
  <div class="col-xs-12">
    <h1 align="center">Hari Menuju Hari Guru Nasional 2024</h1>
    <section id="count-up-section" class="count-up-section">
      <div class="container">

        <div class="row">
          <div class="col-xs-12" style="height: 30vh; display: flex; align-items: center;
justify-content: center;">

            <div class="flipTimemodulesboxes">

              <div class="flipTimebox">
                <div class="flipclock1"></div>
                <div class="flipclock1message"></div>
              </div>

            </div>

          </div>
          <!-- end col -->
        </div>
        <!-- end row -->

      </div>
      <!-- end container -->

    </section>
  </div>
<!-- partial -->
  <script src='https://cdnjs.cloudflare.com/ajax/libs/jquery/2.2.4/jquery.min.js'></script><script
src="./script.js"></script>
<script>
/*line 1070 to change the date countdown: {var dt1 = "08/25/" + dt.getFullYear() + " 00:00:01
am +0000";} */

var Base = function() {};
(Base.extend = function(a, b) {
  "use strict";
  var c = Base.prototype.extend;
  Base._prototyping = !0;
  var d = new this();
  c.call(d, a), (d.base = function() {}), delete Base._prototyping;
  var e = d.constructor,

```



```

f = (d.constructor = function() {
  if (!Base._prototyping)
    if (this._constructing || this.constructor == f)
      (this._constructing = !0),
      e.apply(this, arguments),
      delete this._constructing;
    else if (null !== arguments[0])
      return (arguments[0].extend || c).call(arguments[0], d);
  });
return (
  (f.ancestor = this),
  (f.extend = this.extend),
  (f.forEach = this.forEach),
  (f.implement = this.implement),
  (f.prototype = d),
  (f.toString = this.toString),
  (f.valueOf = function(a) {
    return "object" == a ? f : e.valueOf();
  }),
  c.call(f, b),
  "function" == typeof f.init && f.init(),
  f
);
}),
(Base.prototype = {
  extend: function(a, b) {
    if (arguments.length > 1) {
      var c = this[a];
      if (
        c &&
        "function" == typeof b &&
        (!c.valueOf || c.valueOf() != b.valueOf()) &&
        /\bbase\b/.test(b)
      ) {
        var d = b.valueOf();
        (b = function() {
          var a = this.base || Base.prototype.base;
          this.base = c;
          var b = d.apply(this, arguments);
          return (this.base = a), b;
        }),
        (b.valueOf = function(a) {
          return "object" == a ? b : d;
        }),
        (b.toString = Base.toString);
      }
      this[a] = b;
    } else if (a) {

```

```

var e = Base.prototype.extend;
Base._prototyping ||
  "function" == typeof this ||
  (e = this.extend || e);
for (
  var f = { toSource: null },
      g = ["constructor", "toString", "valueOf"],
      h = Base._prototyping ? 0 : 1;
  (i = g[h++]));

  )
  a[i] != f[i] && e.call(this, i, a[i]);
for (var i in a) f[i] || e.call(this, i, a[i]);
}
return this;
}
}),
(Base = Base.extend(
{
  constructor: function() {
    this.extend(arguments[0]);
  }
},
{
  ancestor: Object,
  version: "1.1",
  forEach: function(a, b, c) {
    for (var d in a) void 0 === this.prototype[d] && b.call(c, a[d], d, a);
  },
  implement: function() {
    for (var a = 0; a < arguments.length; a++)
      "function" == typeof arguments[a]
        ? arguments[a](this.prototype)
        : this.prototype.extend(arguments[a]);
    return this;
  },
  toString: function() {
    return String(this.valueOf());
  }
}
));
var FlipClock;
!(function(a) {
  "use strict";
  (FlipClock = function(a, b, c) {
    return (
      b instanceof Object && b instanceof Date == !1 && ((c = b), (b = 0)),
      new FlipClock.Factory(a, b, c)
    );
  });

```

```

    );
  }},
  (FlipClock.Lang = {}),
  (FlipClock.Base = Base.extend({
    buildDate: "2014-12-12",
    version: "0.7.7",
    constructor: function(b, c) {
      "object" != typeof b && (b = {}),
      "object" != typeof c && (c = {}),
      this.setOptions(a.extend(!0, {}, b, c));
    },
    callback: function(a) {
      if ("function" == typeof a) {
        for (var b = [], c = 1; c <= arguments.length; c++)
          arguments[c] && b.push(arguments[c]);
        a.apply(this, b);
      }
    },
    log: function(a) {
      window.console && console.log && console.log(a);
    },
    getOption: function(a) {
      return this[a] ? this[a] : !1;
    },
    getOptions: function() {
      return this;
    },
    setOption: function(a, b) {
      this[a] = b;
    },
    setOptions: function(a) {
      for (var b in a) "undefined" != typeof a[b] && this.setOption(b, a[b]);
    }
  }));
})(jQuery),
(function(a) {
  "use strict";
  FlipClock.Face = FlipClock.Base.extend({
    autoStart: !0,
    dividers: [],
    factory: !1,
    lists: [],
    constructor: function(a, b) {
      (this.dividers = []),
      (this.lists = []),
      this.base(b),
      (this.factory = a);
    },
  },

```

```

build: function() {
    this.autoStart && this.start();
},
createDivider: function(b, c, d) {
    ("boolean" !== typeof c && c) || ((d = c), (c = b));
    var e = [
        '<span class="' + this.factory.classes.dot + ' top"></span>',
        '<span class="' + this.factory.classes.dot + ' bottom"></span>'
    ].join("");
    d && (e = ""), (b = this.factory.localize(b));
    var f = [
        '<span class="' +
            this.factory.classes.divider +
            " " +
            (c ? c : "").toLowerCase() +
            "'>',
        '<span class="' +
            this.factory.classes.label +
            "'>' +
            (b ? b : "") +
            "</span>",
        e,
        "</span>"
    ],
    g = a(f.join(""));
    return this.dividers.push(g), g;
},
createList: function(a, b) {
    "object" === typeof a && ((b = a), (a = 0));
    var c = new FlipClock.List(this.factory, a, b);
    return this.lists.push(c), c;
},
reset: function() {
    (this.factory.time = new FlipClock.Time(
        this.factory,
        this.factory.original ? Math.round(this.factory.original) : 0,
        { minimumDigits: this.factory.minimumDigits }
    )),
    this.flip(this.factory.original, !1);
},
appendDigitToClock: function(a) {
    a.$el.append(!1);
},
addDigit: function(a) {
    var b = this.createList(a, {
        classes: {
            active: this.factory.classes.active,
            before: this.factory.classes.before,

```

```

        flip: this.factory.classes.flip
    }
});
this.appendDigitToClock(b);
},
start: function() {},
stop: function() {},
autoIncrement: function() {
    this.factory.countdown ? this.decrement() : this.increment();
},
increment: function() {
    this.factory.time.addSecond();
},
decrement: function() {
    0 == this.factory.time.getTimeSeconds()
        ? this.factory.stop()
        : this.factory.time.subSecond();
},
flip: function(b, c) {
    var d = this;
    a.each(b, function(a, b) {
        var e = d.lists[a];
        e ? (c || b == e.digit || e.play(), e.select(b)) : d.addDigit(b);
    });
}
});
})(jQuery),
(function(a) {
    "use strict";
    FlipClock.Factory = FlipClock.Base.extend({
        animationRate: 1e3,
        autoStart: !0,
        callbacks: {
            destroy: !1,
            create: !1,
            init: !1,
            interval: !1,
            start: !1,
            stop: !1,
            reset: !1
        },
        classes: {
            active: "flip-clock-active",
            before: "flip-clock-before",
            divider: "flip-clock-divider",
            dot: "flip-clock-dot",
            label: "flip-clock-label",
            flip: "flip",

```

```

    play: "play",
    wrapper: "flip-clock-wrapper"
  },
  clockFace: "HourlyCounter",
  countdown: !1,
  defaultClockFace: "HourlyCounter",
  defaultLanguage: "english",
  $el: !1,
  face: !0,
  lang: !1,
  language: "english",
  minimumDigits: 0,
  original: !1,
  running: !1,
  time: !1,
  timer: !1,
  $wrapper: !1,
  constructor: function(b, c, d) {
    d || (d = {}),
    (this.lists = []),
    (this.running = !1),
    this.base(d),
    (this.$el = a(b).addClass(this.classes.wrapper)),
    (this.$wrapper = this.$el),
    (this.original = c instanceof Date ? c : c ? Math.round(c) : 0),
    (this.time = new FlipClock.Time(this, this.original, {
      minimumDigits: this.minimumDigits,
      animationRate: this.animationRate
    })),
    (this.timer = new FlipClock.Timer(this, d)),
    this.loadLanguage(this.language),
    this.loadClockFace(this.clockFace, d),
    this.autoStart && this.start();
  },
  loadClockFace: function(a, b) {
    var c,
        d = "Face",
        e = !1;
    return (
      (a = a.ucfirst() + d),
      this.face.stop && (this.stop(), (e = !0)),
      this.$el.html(""),
      (this.time.minimumDigits = this.minimumDigits),
      (c = FlipClock[a]
        ? new FlipClock[a](this, b)
        : new FlipClock[this.defaultClockFace + d](this, b)),
      c.build(),
      (this.face = c),

```

```

        e && this.start(),
        this.face
    );
},
loadLanguage: function(a) {
    var b;
    return (
        (b = FlipClock.Lang[a.ucfirst()]
        ? FlipClock.Lang[a.ucfirst()]
        : FlipClock.Lang[a]
        ? FlipClock.Lang[a]
        : FlipClock.Lang[this.defaultLanguage]),
        (this.lang = b)
    );
},
localize: function(a, b) {
    var c = this.lang;
    if (!a) return null;
    var d = a.toLowerCase();
    return "object" == typeof b && (c = b), c && c[d] ? c[d] : a;
},
start: function(a) {
    var b = this;
    b.running || (b.countdown && !(b.countdown && b.time.time > 0))
        ? b.log("Trying to start timer when countdown already at 0")
        : (b.face.start(b.time),
            b.timer.start(function() {
                b.flip(), "function" == typeof a && a();
            }));
},
stop: function(a) {
    this.face.stop(), this.timer.stop(a);
    for (var b in this.lists)
        this.lists.hasOwnProperty(b) && this.lists[b].stop();
},
reset: function(a) {
    this.timer.reset(a), this.face.reset();
},
setTime: function(a) {
    (this.time.time = a), this.flip(!0);
},
getTime: function(a) {
    return this.time;
},
setCountdown: function(a) {
    var b = this.running;
    (this.countdown = a ? !0 : !1), b && (this.stop(), this.start());
},

```

```

    flip: function(a) {
        this.face.flip(!1, a);
    }
});
})(jQuery),
(function(a) {
    "use strict";
    FlipClock.List = FlipClock.Base.extend({
        digit: 0,
        classes: {
            active: "flip-clock-active",
            before: "flip-clock-before",
            flip: "flip"
        },
        factory: !1,
        $el: !1,
        $obj: !1,
        items: [],
        lastDigit: 0,
        constructor: function(a, b, c) {
            (this.factory = a),
            (this.digit = b),
            (this.lastDigit = b),
            (this.$el = this.createList()),
            (this.$obj = this.$el),
            b > 0 && this.select(b),
            this.factory.$el.append(this.$el);
        },
        select: function(a) {
            if (
                ("undefined" == typeof a ? (a = this.digit) : (this.digit = a),
                this.digit != this.lastDigit)
            ) {
                var b = this.$el
                    .find("." + this.classes.before)
                    .removeClass(this.classes.before);
                this.$el
                    .find("." + this.classes.active)
                    .removeClass(this.classes.active)
                    .addClass(this.classes.before),
                this.appendListItem(this.classes.active, this.digit),
                b.remove(),
                (this.lastDigit = this.digit);
            }
        },
        play: function() {
            this.$el.addClass(this.factory.classes.play);
        },
    },

```



```

stop: function() {
  var a = this;
  setTimeout(function() {
    a.$el.removeClass(a.factory.classes.play);
  }, this.factory.timer.interval);
},
createListItem: function(a, b) {
  return [
    '<li class="' + (a ? a : "") + ">',
    '<a href="#">',
    '<div class="up">',
    '<div class="shadow"></div>',
    '<div class="inn">' + (b ? b : "") + "</div>",
    "</div>",
    '<div class="down">',
    '<div class="shadow"></div>',
    '<div class="inn">' + (b ? b : "") + "</div>",
    "</div>",
    "</a>",
    "</li>"
  ].join("");
},
appendListItem: function(a, b) {
  var c = this.createListItem(a, b);
  this.$el.append(c);
},
createList: function() {
  var b = this.getPrevDigit() ? this.getPrevDigit() : this.digit,
  c = a(
    [
      '<ul class="' +
        this.classes.flip +
        " " +
        (this.factory.running ? this.factory.classes.play : "") +
        ">',
      this.createListItem(this.classes.before, b),
      this.createListItem(this.classes.active, this.digit),
      "</ul>"
    ].join("")
  );
  return c;
},
getNextDigit: function() {
  return 9 == this.digit ? 0 : this.digit + 1;
},
getPrevDigit: function() {
  return 0 == this.digit ? 9 : this.digit - 1;
}
}

```

```

});
})(jQuery),
(function(a) {
    "use strict";
    (String.prototype.ucfirst = function() {
        return this.substr(0, 1).toUpperCase() + this.substr(1);
    }),
    (a.fn.FlipClock = function(b, c) {
        return new FlipClock(a(this), b, c);
    }),
    (a.fn.flipClock = function(b, c) {
        return a.fn.FlipClock(b, c);
    });
})(jQuery),
(function(a) {
    "use strict";
    FlipClock.Time = FlipClock.Base.extend({
        time: 0,
        factory: !1,
        minimumDigits: 0,
        constructor: function(a, b, c) {
            "object" != typeof c && (c = {}),
            c.minimumDigits || (c.minimumDigits = a.minimumDigits),
            this.base(c),
            (this.factory = a),
            b && (this.time = b);
        },
        convertDigitsToArray: function(a) {
            var b = [];
            a = a.toString();
            for (var c = 0; c < a.length; c++) a[c].match(/^\d*$/g) && b.push(a[c]);
            return b;
        },
        digit: function(a) {
            var b = this.toString(),
                c = b.length;
            return b[c - a] ? b[c - a] : !1;
        },
        digitize: function(b) {
            var c = [];
            if (
                (a.each(b, function(a, b) {
                    (b = b.toString()), 1 == b.length && (b = "0" + b);
                    for (var d = 0; d < b.length; d++) c.push(b.charAt(d));
                })),
                c.length > this.minimumDigits && (this.minimumDigits = c.length),
                this.minimumDigits > c.length)
            )

```

```

        for (var d = c.length; d < this.minimumDigits; d++) c.unshift("0");
        return c;
    },
    getDateObject: function() {
        return this.time instanceof Date
            ? this.time
            : new Date(new Date().getTime() + 1e3 * this.getTimeSeconds());
    },
    getDayCounter: function(a) {
        var b = [this.getDays(), this.getHours(!0), this.getMinutes(!0)];
        return a && b.push(this.getSeconds(!0)), this.digitize(b);
    },
    getDays: function(a) {
        var b = this.getTimeSeconds() / 60 / 60 / 24;
        return a && (b %= 7), Math.floor(b);
    },
    getHourCounter: function() {
        var a = this.digitize([
            this.getHours(),
            this.getMinutes(!0),
            this.getSeconds(!0)
        ]);
        return a;
    },
    getHourly: function() {
        return this.getHourCounter();
    },
    getHours: function(a) {
        var b = this.getTimeSeconds() / 60 / 60;
        return a && (b %= 24), Math.floor(b);
    },
    getMilitaryTime: function(a, b) {
        "undefined" == typeof b && (b = !0), a || (a = this.getDateObject());
        var c = [a.getHours(), a.getMinutes()];
        return b === !0 && c.push(a.getSeconds()), this.digitize(c);
    },
    getMinutes: function(a) {
        var b = this.getTimeSeconds() / 60;
        return a && (b %= 60), Math.floor(b);
    },
    getMinuteCounter: function() {
        var a = this.digitize([this.getMinutes(), this.getSeconds(!0)]);
        return a;
    },
    getTimeSeconds: function(a) {
        return (
            a || (a = new Date()),
            this.time instanceof Date

```

```

        ? this.factory.countdown
        ? Math.max(this.time.getTime() / 1e3 - a.getTime() / 1e3, 0)
        : a.getTime() / 1e3 - this.time.getTime() / 1e3
        : this.time
    );
},
getTime: function(a, b) {
    "undefined" == typeof b && (b = !0),
    a || (a = this.getDateObject()),
    console.log(a);
    var c = a.getHours(),
        d = [c > 12 ? c - 12 : 0 === c ? 12 : c, a.getMinutes()];
    return b === !0 && d.push(a.getSeconds()), this.digitize(d);
},
getSeconds: function(a) {
    var b = this.getTimeSeconds();
    return a && (60 == b ? (b = 0) : (b %= 60)), Math.ceil(b);
},
getWeeks: function(a) {
    var b = this.getTimeSeconds() / 60 / 60 / 24 / 7;
    return a && (b %= 52), Math.floor(b);
},
removeLeadingZeros: function(b, c) {
    var d = 0,
        e = [];
    return (
        a.each(c, function(a, f) {
            b > a ? (d += parseInt(c[a], 10)) : e.push(c[a]);
        }),
        0 === d ? e : c
    );
},
addSeconds: function(a) {
    this.time instanceof Date
        ? this.time.setSeconds(this.time.getSeconds() + a)
        : (this.time += a);
},
addSecond: function() {
    this.addSeconds(1);
},
subSeconds: function(a) {
    this.time instanceof Date
        ? this.time.setSeconds(this.time.getSeconds() - a)
        : (this.time -= a);
},
subSecond: function() {
    this.subSeconds(1);
},

```

```

    toString: function() {
        return this.getTimeSeconds().toString();
    }
});
})(jQuery),
(function(a) {
    "use strict";
    FlipClock.Timer = FlipClock.Base.extend({
        callbacks: {
            destroy: !1,
            create: !1,
            init: !1,
            interval: !1,
            start: !1,
            stop: !1,
            reset: !1
        },
        count: 0,
        factory: !1,
        interval: 1e3,
        animationRate: 1e3,
        constructor: function(a, b) {
            this.base(b),
            (this.factory = a),
            this.callback(this.callbacks.init),
            this.callback(this.callbacks.create);
        },
        getElapsed: function() {
            return this.count * this.interval;
        },
        getElapsedTime: function() {
            return new Date(this.time + this.getElapsed());
        },
        reset: function(a) {
            clearInterval(this.timer),
            (this.count = 0),
            this._setInterval(a),
            this.callback(this.callbacks.reset);
        },
        start: function(a) {
            (this.factory.running = !0),
            this._createTimer(a),
            this.callback(this.callbacks.start);
        },
        stop: function(a) {
            (this.factory.running = !1),
            this._clearInterval(a),
            this.callback(this.callbacks.stop),

```

```

        this.callback(a);
    },
    _clearInterval: function() {
        clearInterval(this.timer);
    },
    _createTimer: function(a) {
        this._setInterval(a);
    },
    _destroyTimer: function(a) {
        this._clearInterval(),
        (this.timer = !1),
        this.callback(a),
        this.callback(this.callbacks.destroy);
    },
    _interval: function(a) {
        this.callback(this.callbacks.interval), this.callback(a), this.count++;
    },
    _setInterval: function(a) {
        var b = this;
        b._interval(a),
        (b.timer = setInterval(function() {
            b._interval(a);
        }, this.interval));
    }
    });
})(jQuery),
(function(a) {
    FlipClock.TwentyFourHourClockFace = FlipClock.Face.extend({
        constructor: function(a, b) {
            this.base(a, b);
        },
        build: function(b) {
            var c = this,
                d = this.factory.$el.find("ul");
            this.factory.time.time ||
            ((this.factory.original = new Date()),
            (this.factory.time = new FlipClock.Time(
                this.factory,
                this.factory.original
            )));
            var b = b ? b : this.factory.time.getMilitaryTime(!1, this.showSeconds);
            b.length > d.length &&
            a.each(b, function(a, b) {
                c.createList(b);
            }),
            this.createDivider(),
            this.createDivider(),
            a(this.dividers[0]).insertBefore(

```

```

        this.lists[this.lists.length - 2].$el
    ),
    a(this.dividers[1]).insertBefore(
        this.lists[this.lists.length - 4].$el
    ),
    this.base();
},
flip: function(a, b) {
    this.autoIncrement(),
    (a = a ? a : this.factory.time.getMilitaryTime(!1, this.showSeconds)),
    this.base(a, b);
}
});
})(jQuery),
(function(a) {
    FlipClock.CounterFace = FlipClock.Face.extend({
        shouldAutoIncrement: !1,
        constructor: function(a, b) {
            "object" != typeof b && (b = {}),
            (a.autoStart = b.autoStart ? !0 : !1),
            b.autoStart && (this.shouldAutoIncrement = !0),
            (a.increment = function() {
                (a.countdown = !1), a.setTime(a.getTime().getTimeSeconds() + 1);
            }),
            (a.decrement = function() {
                a.countdown = !0;
                var b = a.getTime().getTimeSeconds();
                b > 0 && a.setTime(b - 1);
            }),
            (a.setValue = function(b) {
                a.setTime(b);
            }),
            (a.setCounter = function(b) {
                a.setTime(b);
            }),
            this.base(a, b);
        },
        build: function() {
            var b = this,
                c = this.factory.$el.find("ul"),
                d = this.factory.getTime().digitize([this.factory.getTime().time]);
            d.length > c.length &&
            a.each(d, function(a, c) {
                var d = b.createList(c);
                d.select(c);
            }),
            a.each(this.lists, function(a, b) {
                b.play();
            });
        }
    });
})(jQuery);

```

```

    }),
    this.base();
  },
  flip: function(a, b) {
    this.shouldAutoIncrement && this.autoIncrement(),
    a ||
    (a = this.factory
      .getTime()
      .digitize([this.factory.getTime().time])),
    this.base(a, b);
  },
  reset: function() {
    (this.factory.time = new FlipClock.Time(
      this.factory,
      this.factory.original ? Math.round(this.factory.original) : 0
    )),
    this.flip();
  }
});
})(jQuery),
(function(a) {
  FlipClock.DailyCounterFace = FlipClock.Face.extend({
    showSeconds: !0,
    constructor: function(a, b) {
      this.base(a, b);
    },
    build: function(b) {
      var c = this,
          d = this.factory.$el.find("ul"),
          e = 0;
      (b = b ? b : this.factory.time.getDayCounter(this.showSeconds)),
      b.length > d.length &&
      a.each(b, function(a, b) {
        c.createList(b);
      }),
      this.showSeconds
      ? a(this.createDivider("Seconds")).insertBefore(
        this.lists[this.lists.length - 2].$el
      )
      : (e = 2),
      a(this.createDivider("Minutes")).insertBefore(
        this.lists[this.lists.length - 4 + e].$el
      ),
      a(this.createDivider("Hours")).insertBefore(
        this.lists[this.lists.length - 6 + e].$el
      ),
      a(this.createDivider("Days", !0)).insertBefore(this.lists[0].$el),
      this.base();
    }
  });

```



```

    },
    flip: function(a, b) {
        a || (a = this.factory.time.getDayCounter(this.showSeconds)),
        this.autoIncrement(),
        this.base(a, b);
    }
});
})(jQuery),
(function(a) {
    FlipClock.HourlyCounterFace = FlipClock.Face.extend({
        constructor: function(a, b) {
            this.base(a, b);
        },
        build: function(b, c) {
            var d = this,
                e = this.factory.$el.find("ul");
            (c = c ? c : this.factory.time.getHourCounter()),
            c.length > e.length &&
            a.each(c, function(a, b) {
                d.createList(b);
            }),
            a(this.createDivider("Seconds")).insertBefore(
                this.lists[this.lists.length - 2].$el
            ),
            a(this.createDivider("Minutes")).insertBefore(
                this.lists[this.lists.length - 4].$el
            ),
            b ||
            a(this.createDivider("Hours", !0)).insertBefore(this.lists[0].$el),
            this.base();
        },
        flip: function(a, b) {
            a || (a = this.factory.time.getHourCounter()),
            this.autoIncrement(),
            this.base(a, b);
        },
        appendDigitToClock: function(a) {
            this.base(a, this.dividers[0].insertAfter(this.dividers[0].next()));
        }
    });
})(jQuery),
(function(a) {
    FlipClock.MinuteCounterFace = FlipClock.HourlyCounterFace.extend({
        clearExcessDigits: !1,
        constructor: function(a, b) {
            this.base(a, b);
        },
        build: function() {

```

```

        this.base(!0, this.factory.time.getMinuteCounter());
    },
    flip: function(a, b) {
        a || (a = this.factory.time.getMinuteCounter()), this.base(a, b);
    }
});
})(jQuery),
(function(a) {
    FlipClock.TwelveHourClockFace = FlipClock.TwentyFourHourClockFace.extend({
        meridium: !1,
        meridiumText: "AM",
        build: function() {
            var b = this.factory.time.getTime(!1, this.showSeconds);
            this.base(b,
                (this.meridiumText = this.getMeridium()),
                (this.meridium = a(
                    [
                        '<ul class="flip-clock-meridium">',
                        "<li>",
                        '<a href="#">' + this.meridiumText + "</a>",
                        "</li>",
                        "</ul>"
                    ].join(""))
                )),
            this.meridium.insertAfter(this.lists[this.lists.length - 1].$el);
        },
        flip: function(a, b) {
            this.meridiumText !== this.getMeridium() &&
                ((this.meridiumText = this.getMeridium()),
                this.meridium.find("a").html(this.meridiumText)),
            this.base(this.factory.time.getTime(!1, this.showSeconds), b);
        },
        getMeridium: function() {
            return new Date().getHours() >= 12 ? "PM" : "AM";
        },
        isPM: function() {
            return "PM" == this.getMeridium() ? !0 : !1;
        },
        isAM: function() {
            return "AM" == this.getMeridium() ? !0 : !1;
        }
    });
})(jQuery),
(function(a) {
    (FlipClock.Lang.Arabic = {
        years: "سنوات",
        months: "شهور",
        days: "أيام",

```

```

    hours: "ساعات",
    minutes: "دقائق",
    seconds: "ثواني"
  }},
  (FlipClock.Lang.ar = FlipClock.Lang.Arabic),
  (FlipClock.Lang["ar-ar"] = FlipClock.Lang.Arabic),
  (FlipClock.Lang.arabic = FlipClock.Lang.Arabic);
})(jQuery),
(function(a) {
  (FlipClock.Lang.Danish = {
    years: "År",
    months: "Måneder",
    days: "Dage",
    hours: "Timer",
    minutes: "Minutter",
    seconds: "Sekunder"
  }),
  (FlipClock.Lang.da = FlipClock.Lang.Danish),
  (FlipClock.Lang["da-dk"] = FlipClock.Lang.Danish),
  (FlipClock.Lang.danish = FlipClock.Lang.Danish);
})(jQuery),
(function(a) {
  (FlipClock.Lang.German = {
    years: "Jahre",
    months: "Monate",
    days: "Tage",
    hours: "Stunden",
    minutes: "Minuten",
    seconds: "Sekunden"
  }),
  (FlipClock.Lang.de = FlipClock.Lang.German),
  (FlipClock.Lang["de-de"] = FlipClock.Lang.German),
  (FlipClock.Lang.german = FlipClock.Lang.German);
})(jQuery),
(function(a) {
  (FlipClock.Lang.English = {
    years: "Years",
    months: "Months",
    days: "Days",
    hours: "Hours",
    minutes: "Minutes",
    seconds: "Seconds"
  }),
  (FlipClock.Lang.en = FlipClock.Lang.English),
  (FlipClock.Lang["en-us"] = FlipClock.Lang.English),
  (FlipClock.Lang.english = FlipClock.Lang.English);
})(jQuery),
(function(a) {

```

```

(FlipClock.Lang.Spanish = {
  years: "Años",
  months: "Meses",
  days: "Días",
  hours: "Horas",
  minutes: "Minutos",
  seconds: "Segundos"
}),
(FlipClock.Lang.es = FlipClock.Lang.Spanish),
(FlipClock.Lang["es-es"] = FlipClock.Lang.Spanish),
(FlipClock.Lang.spanish = FlipClock.Lang.Spanish);
})(jQuery),
(function(a) {
  (FlipClock.Lang.Finnish = {
    years: "Vuotta",
    months: "Kuukautta",
    days: "Päivää",
    hours: "Tuntia",
    minutes: "Minuuttia",
    seconds: "Sekuntia"
  }),
  (FlipClock.Lang.fi = FlipClock.Lang.Finnish),
  (FlipClock.Lang["fi-fi"] = FlipClock.Lang.Finnish),
  (FlipClock.Lang.finnish = FlipClock.Lang.Finnish);
})(jQuery),
(function(a) {
  (FlipClock.Lang.French = {
    years: "Ans",
    months: "Mois",
    days: "Jours",
    hours: "Heures",
    minutes: "Minutes",
    seconds: "Secondes"
  }),
  (FlipClock.Lang.fr = FlipClock.Lang.French),
  (FlipClock.Lang["fr-ca"] = FlipClock.Lang.French),
  (FlipClock.Lang.french = FlipClock.Lang.French);
})(jQuery),
(function(a) {
  (FlipClock.Lang.Italian = {
    years: "Anni",
    months: "Mesi",
    days: "Giorni",
    hours: "Ore",
    minutes: "Minuti",
    seconds: "Secondi"
  }),
  (FlipClock.Lang.it = FlipClock.Lang.Italian),

```

```

        (FlipClock.Lang["it-it"] = FlipClock.Lang.Italian),
        (FlipClock.Lang.italian = FlipClock.Lang.Italian);
    })(jQuery),
    (function(a) {
        (FlipClock.Lang.Latvian = {
            years: "Gadi",
            months: "Mēneši",
            days: "Dienas",
            hours: "Stundas",
            minutes: "Minūtes",
            seconds: "Sekundes"
        }),
        (FlipClock.Lang.lv = FlipClock.Lang.Latvian),
        (FlipClock.Lang["lv-lv"] = FlipClock.Lang.Latvian),
        (FlipClock.Lang.latvian = FlipClock.Lang.Latvian);
    })(jQuery),
    (function(a) {
        (FlipClock.Lang.Dutch = {
            years: "Jaren",
            months: "Maanden",
            days: "Dagen",
            hours: "Uren",
            minutes: "Minuten",
            seconds: "Seconden"
        }),
        (FlipClock.Lang.nl = FlipClock.Lang.Dutch),
        (FlipClock.Lang["nl-be"] = FlipClock.Lang.Dutch),
        (FlipClock.Lang.dutch = FlipClock.Lang.Dutch);
    })(jQuery),
    (function(a) {
        (FlipClock.Lang.Norwegian = {
            years: "År",
            months: "Måneder",
            days: "Dager",
            hours: "Timer",
            minutes: "Minutter",
            seconds: "Sekunder"
        }),
        (FlipClock.Lang.no = FlipClock.Lang.Norwegian),
        (FlipClock.Lang.nb = FlipClock.Lang.Norwegian),
        (FlipClock.Lang["no-nb"] = FlipClock.Lang.Norwegian),
        (FlipClock.Lang.norwegian = FlipClock.Lang.Norwegian);
    })(jQuery),
    (function(a) {
        (FlipClock.Lang.Portuguese = {
            years: "Anos",
            months: "Meses",
            days: "Dias",

```

```

    hours: "Horas",
    minutes: "Minutos",
    seconds: "Segundos"
  }},
  (FlipClock.Lang.pt = FlipClock.Lang.Portuguese),
  (FlipClock.Lang["pt-br"] = FlipClock.Lang.Portuguese),
  (FlipClock.Lang.portuguese = FlipClock.Lang.Portuguese);
})(jQuery),
(function(a) {
  (FlipClock.Lang.Russian = {
    years: "лет",
    months: "месяцев",
    days: "дней",
    hours: "часов",
    minutes: "минут",
    seconds: "секунд"
  }},
  (FlipClock.Lang.ru = FlipClock.Lang.Russian),
  (FlipClock.Lang["ru-ru"] = FlipClock.Lang.Russian),
  (FlipClock.Lang.russian = FlipClock.Lang.Russian);
})(jQuery),
(function(a) {
  (FlipClock.Lang.Swedish = {
    years: "År",
    months: "Månader",
    days: "Dagar",
    hours: "Timmar",
    minutes: "Minuter",
    seconds: "Sekunder"
  }},
  (FlipClock.Lang.sv = FlipClock.Lang.Swedish),
  (FlipClock.Lang["sv-se"] = FlipClock.Lang.Swedish),
  (FlipClock.Lang.swedish = FlipClock.Lang.Swedish);
})(jQuery),
(function(a) {
  (FlipClock.Lang.Chinese = {
    years: "年",
    months: "月",
    days: "日",
    hours: "时",
    minutes: "分",
    seconds: "秒"
  }},
  (FlipClock.Lang.zh = FlipClock.Lang.Chinese),
  (FlipClock.Lang["zh-cn"] = FlipClock.Lang.Chinese),
  (FlipClock.Lang.chinese = FlipClock.Lang.Chinese);
})(jQuery);
/*change var dt1 date to end date*/

```

```

document.addEventListener("touchstart", function() {}, false);
(function($) {
    "use strict";
    var dt = new Date();
    var cts = Math.ceil(new Date().getTime() / 1000);
    var dt1 = "11/25/" + dt.getFullYear() + " 00:00:01 am +0000";
    var dtClock1 = Math.ceil(new Date(dt1).getTime() / 1000);
    var flipTimeboxSeconds1 = Math.ceil(dtClock1 - cts);
    var dt2 = "01/01/" + (dt.getFullYear() + 1) + " 00:00:01 am +0000";
    var dtClock2 = Math.ceil(new Date(dt2).getTime() / 1000);
    var flipTimeboxSeconds2 = Math.ceil(dtClock2 - cts);
    var flipTimeboxSeconds24hours = 3600 * 24;
    var flipclock1;
    var opts1 = {
        clockFace: "DailyCounter",
        countdown: true,
        callbacks: {
            stop: function() {
                $(".flipclock1message").html("");
            }
        }
    };
    $(".flipclock1").FlipClock(flipTimeboxSeconds1, opts1);
    var flipclock2;
    var opts2 = {
        clockFace: "DailyCounter",
        countdown: true,
        callbacks: {
            stop: function() {
                $(".flipclock2message").html("Time Up! It is Time to go Live!!!");
            }
        }
    };
    $(".flipclock2").FlipClock(flipTimeboxSeconds2, opts2);
    var opts3 = {
        clockFace: "DailyCounter",
        countdown: true,
        callbacks: {
            stop: function() {
                $(".flipclock3message").html("Time Up! It is Time to go Live!!!");
            }
        }
    };
    $(".flipclock3").FlipClock(flipTimeboxSeconds2, opts3);
    var opts4 = {
        clockFace: "DailyCounter",
        countdown: true,
        callbacks: {

```

```

        stop: function() {
            $(".flipclock4message").html("Time Up! It is Time to go Live!!!");
        }
    };
    $(".flipclock4").FlipClock(flipTimeboxSeconds2, opts4);
    FlipClock.Lang.Custom = {
        days: "Jours",
        hours: "Heures",
        minutes: "Minutes",
        seconds: "Secondes"
    };
    var opts5 = {
        clockFace: "DailyCounter",
        countdown: true,
        language: "Custom",
        callbacks: {
            stop: function() {
                $(".flipclock5message").html("Time Up! It is Time to go Live!!!");
            }
        }
    };
    $(".flipclock5").FlipClock(flipTimeboxSeconds1, opts5);
    var flipclock6;
    var opts6 = {
        clockFace: "HourCounter",
        countdown: true,
        callbacks: {
            stop: function() {
                $(".flipclock6message").html("Time Up! It is Time to go Live!!!");
            }
        }
    };
    $(".flipclock6").FlipClock(flipTimeboxSeconds2, opts6);
    var flipclock7;
    var opts7 = {
        clockFace: "HourCounter",
        countdown: true,
        callbacks: {
            stop: function() {
                $(".flipclock7message").html("Time Up! It is Time to go Live!!!");
            }
        }
    };
    $(".flipclock7").FlipClock(flipTimeboxSeconds2, opts7);
    var flipclock8;
    var opts8 = {
        clockFace: "HourCounter",

```



```

    countdown: true,
    callbacks: {
      stop: function() {
        $(".flipclock8message").html("Time Up! It is Time to go Live!!!");
      }
    }
  };
  $(".flipclock8").FlipClock(flipTimeboxSeconds24hours, opts8);
  var flipclock9;
  var opts9 = { clockFace: "TwelveHourClock" };
  $(".flipclock9").FlipClock(opts9);
  var flipclock10;
  var opts10 = { clockFace: "TwelveHourClock" };
  $(".flipclock10").FlipClock(opts10);
  var flipclock11;
  var opts11 = { clockFace: "TwelveHourClock" };
  $(".flipclock11").FlipClock(opts11);
  var flipclock12;
  var opts12 = { clockFace: "TwentyFourHourClock" };
  $(".flipclock12").FlipClock(opts12);
  var flipclock13;
  var opts13 = { clockFace: "MinuteCounter" };
  $(".flipclock13").FlipClock(opts13);
  var flipclock14;
  var opts14 = { clockFace: "MinuteCounter", countdown: true };
  $(".flipclock14").FlipClock(3600, opts14);
  var flipclock15;
  var opts15 = { clockFace: "MinuteCounter", countdown: true };
  $(".flipclock15").FlipClock(1800, opts15);
  var flipclock16 = $(".flipclock16").FlipClock(0, { clockFace: "Counter" });
  setTimeout(function() {
    setInterval(function() {
      flipclock16.increment();
    }, 1000);
  });
  var flipclock17 = $(".flipclock17").FlipClock(0, { clockFace: "Counter" });
  setTimeout(function() {
    setInterval(function() {
      flipclock17.increment();
    }, 1000);
  });
  var flipclock18 = $(".flipclock18").FlipClock(500, {
    clockFace: "Counter",
    countdown: true
  });
  setTimeout(function() {
    setInterval(function() {
      flipclock18.increment();
    }, 1000);
  });

```

```
    }, 1000);
  });
  var flipclock19;
  var opts19 = {
    clockFace: "DailyCounter",
    countdown: true,
    callbacks: {
      stop: function() {
        $(".flipclock19message").html("Time Up! It is Time to go Live!!!");
      }
    }
  };
  $(".flipclock19").FlipClock(flipTimeboxSeconds2, opts19);
  var flipclock20;
  var opts20 = {
    clockFace: "DailyCounter",
    countdown: true,
    callbacks: {
      stop: function() {
        $(".flipclock20message").html("Time Up! It is Time to go Live!!!");
      }
    }
  };
  $(".flipclock20").FlipClock(flipTimeboxSeconds2, opts20);
})(jQuery);
</script>
</body>
</html>
```