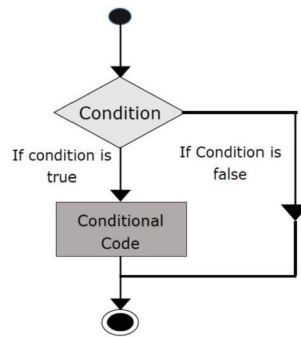




Project: Conditionals I

On command line, create a file called 'conditionals_project.js' in your *ASCProjects/week1/day3* directory.

Part 1 - CONDITIONALS GYMNASTICS

Task:

Predict the output of the following programs.

Program #1:

```
let boba = 5;

if (boba < 3) {
  console.log("!");
}

if (boba == 5) {
  console.log("@");
}

if (boba <= 5) {
  console.log("#");
}

if (boba > 5) {
  console.log("%");
}
```

Program #2:

```
let boba = 5;

if (boba < 3) {
  console.log("!");
}

else if (boba == 5) {
  console.log("@");
}

else if (boba <= 5) {
  console.log("#");
}

else if (boba > 5) {
  console.log("%");
}
```

Program #3:

```
let boba = 5;

if (boba < 3) {
  console.log("!");
}

else if (boba == 5) {
  console.log("@");
}

else if (boba <= 5) {
  console.log("#");
}

if (boba > 5) {
  console.log("%");
}

else {
  console.log("^");
}
```

Program #4:

```
let count = 0;

if (1 > -1) {
  count += 5;
  count *= 2;
}

if ('a' == 'A') {
  count = count - 2;
  count /= 4;
}

if ('apple' > 'alien') {
  count--;
  count--;
  count--;
}

console.log(count);
```

Program #5:

```
if (true) {
  console.log("true blocks are executed");
}

if (false) {
  console.log("false blocks are NOT executed");
}
```

Part 2 - GUESS THAT NUMBER

Task:

Write a program that **generates a random number** between 0 and 100 (inclusive) every time the program runs. The program would **take in a user input** (user's guess), **compare it to the random number** generated, and **output** if the user guessed **right**, **too low**, or **too high**.

**You can safely assume that the user will always enter a numeric value between 0 and 100 (inclusive).*

Sample runs:

User Input: 10

```
ASCStudent@DESKTOP-RLIDLQU MINGW64 ~/Desktop/ASCProjects/week1/day2
$ node conditionals_project.js 10
Your guess: 10
Random number generated: 48
Your guess is TOO LOW
```

User Input: 85

```
ASCStudent@DESKTOP-RLIDLQU MINGW64 ~/Desktop/ASCProjects/week1/day2
$ node conditionals_project.js 85
Your guess: 85
Random number generated: 31
Your guess is TOO HIGH
```

User Input: 30

```
ASCStudent@DESKTOP-RLIDLQU MINGW64 ~/Desktop/ASCProjects/week1/day2
$ node conditionals_project.js 30
Your guess: 30
Random number generated: 30
YOU GUESSED RIGHT!!!
```

Part 3 - GRADER BOT

Task:

Write a program that **takes in user input**, **computes score**, and **outputs varied responses** based on the user input.

- The program begins with an **initial score of 0**

User Input	Score	Output
A	Minus 1	Wrong answer :(
B	Minus 1	Wrong answer :(
C	Plus 5	RIGHT ANSWER! WOHO!!!
D	Minus 50	Bad bad bad...
I.KNOW.THE.SECRET	Plus 1,000,000	Welcome to NIRVANA
(everything else)	No change	Invalid response...

Sample runs:

User Input: A

```
ASCStudent@DESKTOP-RLIDLQU MINGW64 ~/Desktop/ASCProjects/week1/day2
$ node conditionals_project.js A
Initial score: 0

Wrong answer :(
Final score: -1
```

User Input: C

```
ASCStudent@DESKTOP-RLIDLQU MINGW64 ~/Desktop/ASCProjects/week1/day2
$ node conditionals_project.js C
Initial score: 0

RIGHT ANSWER! WOHO!!!
Final score: 5
```

User Input: Wassupppp?

```
ASCStudent@DESKTOP-RLIDLQU MINGW64 ~/Desktop/ASCProjects/week1/day2
$ node conditionals_project.js Wassupppp?
Initial score: 0

Invalid response...
Final score: 0
```

User Input: I.KNOW.THE.SECRET

```
ASCStudent@DESKTOP-RLIDLQU MINGW64 ~/Desktop/ASCProjects/week1/day2
$ node conditionals_project.js I.KNOW.THE.SECRET
Initial score: 0

Welcome to NIRVANA
Final score: 1000000
```

Going Beyond (This part is not required, but you're encouraged to try it!):

Modify your program to take into account **empty user responses**.

User Input	Score	Output
(none)	No change	Empty response... Bad user

Sample run:

User Input: (none)

```
ASCStudent@DESKTOP-RLIDLQU MINGW64 ~/Desktop/ASCProjects/week1/day2
$ node conditionals_project.js
Initial score: 0

Empty response... Bad user
Final score: 0
```

Extra Credit - WHO'S THAT POKEMON?

Task:

Write a program that takes in **two user inputs** - **species** and **name** and outputs info based on the two user inputs.

Note:

Nested conditional statements will be a great tool for this program.

species	name	Output
Pokemon	Pikachu	Hello Pikachu, you're an electric mouse!
	Charmander	Hello Charmander, your final evolution is cool beans!
	(everything else)	Hello <i>(everything else)</i> , we do not have your Pokemon data yet...
human	(input)	Hello <i>(input)</i> , you're a human, not a Pokemon :(
(everything else)	(input)	Unknown species... INTRUDER INTRUDER!

Sample runs:

User Input: Pokemon Pikachu

```
ASCStudent@DESKTOP-RLIDLQU MINGW64 ~/Desktop/ASCProjects/week1/day2
$ node conditionals_project.js Pokemon Pikachu
Hello Pikachu, you're an electric mouse!
```

User Input: Pokemon Mewtwo

```
ASCStudent@DESKTOP-RLIDLQU MINGW64 ~/Desktop/ASCProjects/week1/day2
$ node conditionals_project.js Pokemon Mewtwo
Hello Mewtwo, we do not have your Pokemon data yet...
```

User Input: human Khye

```
ASCStudent@DESKTOP-RLIDLQU MINGW64 ~/Desktop/ASCProjects/week1/day2
$ node conditionals_project.js human Khye
Hello Khye, you're a human, not a Pokemon :(
```

User Input: superhero Aquaman

```
ASCStudent@DESKTOP-RLIDLQU MINGW64 ~/Desktop/ASCProjects/week1/day2
$ node conditionals_project.js superhero Aquaman
Unknown species... INTRUDER INTRUDER!
```