Girls Who Code Club at Foster City Library - every Monday @ 6:30 - 8:30 pm

Oct 2, 2017

- Ice breakers

Oct 10, 2017

Basics of Computer Science

-bit: smallest unit of storage for a computer(either 1 or 0)

Download PyCharm Community Ver

Python interpreter in terminal:

Character: 'c'

Integer: 3

String: "Hello world"

Declare variable: x = "hello world"

Boolean logic: true or false -- executes if true

-Ex:

```
x = 10
>>> x <= 14
True
>>> x>= 14
False
>>> x == 14
False
>>> x != 14
True
>>> x < 14
True
>>> x > 14
False
```

-Ex:

```
>>> True or True
True
>>> True or False
True
>>> False or False
False
>>> False and False
False
```

If else statements:

- If you have a condition and that condition is satisfied, execute
- If condition is not satisfied, execute else
- Ex:
- if x is 30:

- print(x)
- elif y is 2:
- print(y)
- else:
- print("no")

Data Structures

List:

- Ex:
- listC = [x,y,z,"hello"]
- >>>listC
- [10, 10, 'sdsjd', 'hello']
- Index of syntax: listC[0] >>> 10
- Access list in a list: listC[0][5]
- Append
- listB
- [10, 10, 'sdsjd', 'hello', 'dog']
- >>> listB.append("strawberry")
- >>> listB
- [10, 10, 'sdsjd', 'hello', 'dog', 'strawberry']
- Length
- len(listB)
- 6
- Pop(store element of a list in a new variable)
- fruit = listB.pop(5)
- >>> fruit
- 'strawberry'
- >>> listB
- [10, 10, 'sdsjd', 'hello', 'dog']
- Delete
- listB
- [10, 10, 'sdsjd', 'hello', 'dog']
- >>> del listB[1]
- >>> listB
- [10, 'sdsjd', 'hello', 'dog']
-

DIctionary

- Ex:
- phonebook = {"harry": 10,"hermione": 5,"ron": 0}
- >>> phonebook["hermione"]
- 5
- ---'harry' would be the key and 10 would be the value
- Add/replace key(unique identifier):
- phonebook["trevor"] = 174

Oct 16, 2017

Li Fei-Fei video

2 types of loops: for loops and while loops

For loop:

```
for coat in prices_of_coats:
  print(coat)
```

While loop:

```
i = 0
sum2 = 0
while i < len(prices_of_coats):
  sum2 = sum2 + prices_of_coats[i]
  print (sum2)
  i = i + 1
```

Combining a list of strings with while loop:

```
i = 0
str = ""
list_a = ["the", "snake", "ate", "the", "elephant"]
while i < len(list_a):
  str += list_a[i] + " "
  i = i + 1
print(str)
```

Combining a list of strings with for loop:

```
str = ""
for element in list_a:
  str = str + element + " "
print (str)
```

List Indexing(zero based indexing):

```
list_a = ["the", "snake", "ate", "the", "elephant"]
           0        1       2       3         4
len(list_a) --> 5
```

Oct 23, 2017

If/else recap

```
x = 10
if(x==9):
  print("x is 10")
else:
  print("x is not 10")
```

>>x is not 10

Iterate through list with for loop

```
food = ["candy", "chocolate", "yogurt", "pasta"]
for stuff in food:
  if stuff == "yogurt":
    print(stuff)
```

Count Even and Odd numbers in a list

```
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9]
```

```
count_odd = 0
count_even = 0
for x in numbers:
  if x % 2 == 0:
    count_even += 1
  else:
    count_odd += 1
print("Number of even numbers: " + str(count_even))
print("Number of odd numbers: " + str(count_odd))
```

%-modulo(mod): the remainder in long division
Ex: 9%2 is 1

str(): whatever you put in between parentheses turns into a string

Input from user

```
stuff = raw_input("Input some input: ")
print (stuff)
```

Convert b/w Fahrenheit and Celsius

```
scaleInitial = raw_input("Fahrenheit or Celsius? ")
tempInitial = raw_input("What is the original temperature? ")
if(scaleInitial == "Fahrenheit" or scaleInitial == "F"):
  tempFinal = (int(tempInitial)-32 * (5/9))
  print("Final Temperature: ", tempFinal)
elif(scaleInitial == "Celsius" or scaleInitial == "C"):
  tempFinal = (int(tempInitial)*(9/5)) + 32
  print("Final Temperature: ", tempFinal)
else:
  print("Does not compute.")
```

Watched Erica Kochi Video

Oct 30, 2017
Functions(have open parenthesis, condense code):
print
raw_input

```
list_3 = [1, 2, 4, 5, 10]
def print_item(an_item):    ------> you use def to start a function, what's inside parenthesis is input variable
  print(an_item)

print_item(list_3)   ----> calls function and what's inside the parenthesis is the input into the function
```

Ex:

```
# output: [or all odds] or "3, 5, 7"
```

```
def find_odd(lower_bound, upper_bound):
  arr = []
  while lower_bound < upper_bound:
    if lower_bound % 2 != 0:
      arr.append(lower_bound)
    lower_bound += 1
  print arr

find_odd(2, 8)
```

```
print range(0,10)
```
>>>[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

Ex:

```
def get_file_name(lower_bound_1, lower_bound_2, upper_bound_1, upper_bound_2):
  arr1 = range(lower_bound_1, upper_bound_1+1)
  arr2 = range(lower_bound_2, upper_bound_2+1)
  for num1 in arr1:
    for num2 in arr2:
      print("file-"+str(num1)+"-png-"+str(num2))

get_file_name(1, 10, 2, 12)

#output
"file-1-png-10"
"file-1-png-11"
"file-1-png-12"
"file-2-png-10"
"file-2-png-11"
"file-2-png-12"
```

Nov 7

Bret and Rebecca came to talk about Sumo Logic
-enterprise app to troubleshoot
Product Design
UX Design
-work closely with engineers
Design Process
-identify problems
-explore
-design ←
-create |
-try it out |
-make it better --

Nov 14

Bank Account

```
#password verification
#withdrawal
#deposit

#bank account dictionaries
monies = {0: 100,
        1: 30.45,
        2: 9000}

user_names = {0: "Mahak",
            1: "Kelly",
            2: "Isabel"}

user_passwords = {0: "steph",
                1: "curry",
                2: "30"}

#deposit
def deposit(amount_to_deposit, account_num):
  monies[account_num] += amount_to_deposit
  return monies[account_num]


acct = int(raw_input("What is you account number?"))
to_deposit = int(raw_input("What is the $$$$ you wish to deposit"))
print(deposit(to_deposit, acct))

#withdraw
def withdrawl(amount_to_withdraw, account_num):
  if amount_to_withdraw > monies[account_num]:
     print("NOT ENOUGH MONIES")
  else:
     monies[account_num] -= amount_to_withdraw
  return monies[account_num]

acct1 = int(raw_input("What is you account number?"))
to_withdraw = int(raw_input("What is the $$$$ you wish to withdraw"))
print(withdrawl(to_withdraw, acct1))

#login
def login(user_name, password, account_num):
  if user_names[account_num] == user_name and user_passwords[account_num] == password:
     print("Welcome")
  else:
     print("Not welcome")


username = raw_input("What is your username?")
```

```
password = raw_input("What is you password?")
acct_num = int(raw_input("What is your account number?"))
print(login(username, password, acct_num))
```

Nov 27
CSS - how things look, format
https://docs.google.com/presentation/d/1F_CTs9Xjuu09RR2iXVoHlgTDbr0mkgAV5KXs-fygtGM/edit#slide=id.gf5daffb34_0_0
https://www.w3schools.com
https://html5up.net

January 16
Brainstorm semester project ideas
Look for competitors and what they need improvement on

January 22
Format projects
Similar to when UX designers came, draw out potential project page design/template on paper
Create potential survey for project for consumer feedback and guage consumer on how much they like project
Activity sets in Girls Who Code HQ

January 29
Javascript
-created to make web pages alive
-creates fuctionality of web page
-allows web page to respond to user input
-can execute anywhere with a JS engine(JS virtual machine)
JS editor: https://repl.it
CODE:
Strings
'YAY!'
=> 'YAY!'
"There's a four day weekend coming up";
=> "There is a four day weekend coming up"
console.log('Hello'); → aka print in java
Hello
=> undefined
'It\'s';
=> 'it\'s'
JS Methods: https://developer.mozilla.org/en-US/docs/WEB/JAVASCRIPT

"Museum".length;
=> 6
"WHY".toLowerCase();
=> 'why'

-JS doesnt destinguish between whole numbers and decimals
-fractions dont exist
10 + 3452.24357472685472482
=> 3462.2435747268546

-booleans: true or false
Store boolean in variables:
var kitchenLightsOn = false;
=> undefined
console.log(kitchenLightsOn);
false
=> undefined
kitchenLightsOn = true;
console.log(kitchenLightsOn);
true
=> undefined

Operators
-concatenation: add strings
"news" + "paper"
=> 'newspaper'

Variables
→ named values and can store any type of JS value
How to declare variable:
var x = 100; → can access anywhere
let y = 200; → scope issue; if you have two let variables with same name and in different functions they have different scopes
const acceleration = 9.8; → constant variable, some value you dont want to change

Difference with var and let:
```
function varTest(){
  var x = 1;
  if(true){
    var x = 2;
    console.log(x);  // 2 is output
  }
  console.log(x);  // 2 is output
}
```
→ reassigning value of x

```
function letTest(){
  let x = 1;
  if(true){
    let x = 2;          // only exists inside if statements
    console.log(x);  // value of x is 2
  }
  console.log(x);  // value of x is 1
}
```

Frontend: everything you see and do on let's say facebook, ex: clicking login button; designing
Backend: what web servors do; the code operating

Clarify semester project plans

February 12

Javascript

Create arrays

```
var y = [10, 11, "sdkjffjksd"];
console.log(y);
>>[ 10, 11, 'sdkjffjksd' ]
var y = [10, 11, "sdkjffjksd"];
console.log(y[1]);
>>11
```

Pop

```
var y = [10, 11, "sdkjffjksd"];
y.pop();
console.log(y);
>>[ 10, 11 ]
```

Push

```
var y = [10, 11, "sdkjffjksd"];
y.push("orange");
console.log(y);
>>[ 10, 11, 'sdkjffjksd', 'orange' ]
```

IndexOf

```
var y = [10, 11, "sdkjffjksd"];
var x = y.indexOf(11);
console.log(x);
>>1
```

Python

```
y = ["cat", 10, 10.0]
```

```
z = ["dog", 9, 8]
z.insert(2, y[2])
print(z)
>>['dog', 9, 10.0, 8]
```

JavaScript

```
var y = ["cat", 10, 10.0];
var z = ["dog", 9, 8];
var y_Elem = y.pop();
z.push(y_Elem);
console.log(z);
>>['dog', 9, 10.0, 8]
```

For Loops

```
for(var x = 0; x <= 5; x++){
  console.log(x);
}
>>
0
1
2
3
4
5
let y = 1000;
var accelerations = [9.8, 10, 5, 4];
for(var x = 0; x <= 3; x++){
  console.log("x =", x);
  let y = accelerations[x];
  console.log(y);
}
console.log(y);
>>
x = 0
9.8
x = 1
10
x = 2
5
x = 3
4
1000
var accelerations = [9.8, 10, 5, 4];
var snap = [18, 20, 1, -1];
var sum = 0
```

```
for(var x = 0; x <= 3; x++){
  sum += accelerations[x] + snap[x];
}
console.log(sum);
>>66.8
var accelerations = [9.8, 10, 5, 4];
var snap = [18, 20, 1, -1];
var words = ["banana", "cucumber", "potato"];
var string = "";
for(var x = 0; x <= 2; x++){
  string += words[x];
}
console.log(string);
>>bananacucumberpotato
var numbers = [5, 6, 7];
var words = ["banana", "cucumber", "potato"];
var string = "I have ";
for(var x = 0; x < words.length; x++){
  string += numbers[x] + " " + words[x];
  if(x != words.length - 1){
    string += ", ";
  }else{
    string += "."
  }
}
console.log(string);
>>I have 5 banana, 6 cucumber, 7 potato.
```

February 26

Javascript Functions

Functions- block of code that can be named and reused

Return - keyword that exits a function and shares an optional value outside

Ex:

Method 1:

```
function addTwoNumbers(num1, num2){
        return num1 + num2;
}
addTwoNumbers(23, 3);
=> 3
```

Method 2:

```
let addTwoNumbers = function (x,y){
        x + y;
}
```

Pro tip: //plan code before coding ;)

EX: multiply each value in an array by a number

```
//arr and num are parameters
var arr = [2, 3, 12, 5];
var num = 2;
function arrMultiplier(arr, num){
  //creating the variable array result to store the outcome
  let result = [];
  //for loop to iterate through arr to multiply elements in array by number
  for(let i = 0; i < arr.length; i++){
    result.push(arr[i]*num);
  }
  return result;
}
//calls function
arrMultiplier(arr,num);
```

=> [4, 6, 24, 10]

Conditionals

Controls behavior and determines whether or not pieces of the code can run

```
function oddNumber(number){
        if(number % 2 == 0){
          return number + ' is an even number';
  }else{
          return number + ' is an odd number';
  }
}
oddNumber(3);
```

=> '3 is an odd number'

Else if vs Switch

Else if:

```
function greetings(hour){
  let time = hour;
  let greeting;
  if(time < 10){
    greeting = "Good Morning";
  }else if(time < 20){
    greeting = "Good Day";
  }else{
    greeting = "Good Evening";
  }
  return greeting;
}
```

```
greetings(3);
=> 'Good Morning'
```

Switch:

```
function tellMeTheDay(){
  switch (new Date().getDay()){ // creates new date and gives day
    case 6:   //each case is a new condition, 6 would be the condition
      text = "Today is Saturday";
        break;    //break is to escape case once the condition is satisfied
    case 0:
      text = "Today is Sunday";
        break;
    default:    //what happens when cases aren't satisfied
      text = "Looking forward to next weekend";
  }
  return text;
}
tellMeTheDay();
=> 'Looking forward to next weekend'
```

Array

Some functions:

-concat

-push-adds element in array and returns array's length

-pop-removes last element in array

-reverse-returns arrays in reverse order

Objects

-they are values that can contain other values

-they use keys to name values

To get objects key you can:

1.Use dot notation-

objectName.propertyName

2.Use brackets-

objectName[“propertyName”]

EX:

object:

```
var person = {
  firstName: "John",
  lastName: "Doe",
  age:50,
  eyeColor:"blue"
};
```

Calling objects:

```
person.age;
=> 50
Changing key:
person.firstName = "Mathew";
=> 'Mathew'
Add new key/property:
person.married = true;
console.log(person);
>>
{ firstName: 'Mathew',
  lastName: 'Doe',
  age: 50,
  eyeColor: 'blue',
  married: true }

Exercises:
1.
function calcCircumference(radius){
  let circumference = radius * 2 * Math.PI;
  return "The circumference is "+ circumference;
}
calcCircumference(1);
=> 'The circumference is 6.283185307179586'
2.
function calcArea(radius){
  let area = radius * radius * Math.PI;
  return "The area is " + area;
}
calcArea(1);
=> 'The area is 3.141592653589793'
3.
function celsiusToFarenheit(celsius){
  let farenheit = celsius * 1.8 + 32;
  return celsius + "C is " + farenheit + "F";
}
celsiusToFarenheit(5);
=> '5C is 41F'
4.
function farenheitToCelsius(farenheit){
  let celsius = (farenheit - 32)/1.8;
  return farenheit + "F is " + celsius + "C";
}
farenheitToCelsius(5);
=> '5C is 41F'
```

March 5
Review functions in JavaScript
Worked on project

March 12
Worked on project