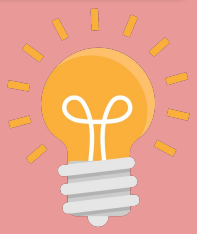
BACK to cover



Unit 3: Making Decisions

3.1 - 3.3: 10/8/2024

⬅️ BACK to divider

Note to self: In Unit 1, I learned concatenation, variable data types, and printing which carried over to Unit 2. In Unit 2, I learned mathematical operators, advanced operations using the code (import math), single line comments (#), multi-line comments (“”), the history of the computer, the random module which finds random numbers between set intervals, the min() & max() function that find the minimum or maximum of a set of values, and the simpleplot module which is used to display data. My only question that I still have is how is python used with HTML and CSS. I should practice building math equations in python more. In Unit 3, we’ve learned to define functions and debug them based on function rules.

*A function is a named group of code that is designed to do a specific task. We have been using functions like print() and input() which are built-in to Python. We can also add modules like math and simpleplot in order to access even more functions.
*Abstraction is the process of reducing complexity by hiding all unnecessary information and focusing on just what is needed to make it work.
*Functions are called by using their name, like print() and input(), and they can be sent information in between in the parentheses. The values entered between the parentheses, separated by commas, are parameters.

- if statements: a structure used to determine if a condition is true or false
  - If true, the block of code is executed
  - If false, the block of code is skipped
- The general syntax of an **if** statement is:

```
if condition:
    code to execute if condition is true
    code to execute if condition is true

code for the rest of your program
```

Overview of Function Syntax

```
def mult_function(x,y):
    z = x * y
    print(“I heart math”)
    return z

mult_function(5,4)
```

Function name
colon
() with parameters inside

Example of built-in functions is abs

- may or may not have inputs (called “arguments”)
- may or may not have outputs (“returns a value”)

Relational Operator	Meaning
<	Less than
>	Greater than
==	Equal to
!=	Not equal to
<=	Less than or equal to
>=	Greater than or equal to

To-do:

✅ **Find ideas for portfolio**
✅ **Get up to 3.5**

Logical Operator	Allows us to check more than one condition at a time
Or	Requires only one expression to be true for the overall condition to be true
And	Requires both conditions to be true for the overall condition to be true
Not	Evaluates the condition and then does the opposite

3.4 - 3.6: 10/17/2024

⬅️ BACK to divider

If-else statement example:

```
if condition:
    print("Condition is true")
else:
    print("Condition is false")
print("This is outside of the if-else-statement. Will print either way")
```

Else-If (Elif) Improper Example:

```
score = int(input("Enter your score: "))

if score >= 1000:
    print("You got three stars.")

if score >= 750 and score < 1000:
    print("You got two stars.")

if score >= 500 and score < 750:
    print("You got one star.")

if score < 500:
    print("You got zero stars.")
```

Else-If (Elif) Proper Example:

```
score = int(input(“Enter your score: “))

if score >= 1000:
  print(“You got three stars.”)
elif score >= 750:
  print(“You got two stars.”)
elif score >= 500:
  print(“You got one star.”)
else:
  print(“You got zero stars.”)
```

Algorithm = a set of rules or instructions for solving or accomplishing a task
Algorithm Guidelines

- Need to be in a specific order
- They need to be clear and precise. Applying all the syntax rules we know in a program will make our instructions clear.
- Should produce a desired result. Without a result, an algorithm is pointless.
- Should be able to run in a finite amount of time.

Speed, memory use, scalability, security, and maintenance make one algorithm better than another.

To-do:

- ✅ Find questions for the Chatbot assignment
- ❑ Update images on Portfolio
- ❑ Talk to Ms.Ramirez about Test 2 grade
- ❑ Finish flowchart

Links to Helpful Documents:

- Unit 3 Vocab
- Python Error Troubleshooting Cheatsheet

Name: Micah Kennedy

Block: _____ Date: 10/24/2024

Unit 3: Test Review

1. What is an if-statement used for?

   Runs a block of code when a condition is met

2. When do you use an else?

   When you need something to happen when the if condition is false

3. What does != mean?

   not equal to

Write an if statement for each of the following:

4. Test if num is between 78 and 45 inclusive.

```
if num >= 45 and num <= 78:
```

5. Test if x is NOT between 67 and 32.

```
if x  <=  32 or x >= 67:
```

6. Test if value is positive.

```
if x > 0:
```

7. Input two numbers and print the smaller to the screen.

```
a = float(input("Enter a number: "))
b = float(input("Enter another number: "))
if a > b:
   print(b)
else:
   print(a)
```

Correct the mistakes (there may be more than one):

8. IF (x < y < 8):
   if x < y and y <8:

9. IF (word = "pumpernickel):
   if word == "pumpernickel":

Answer the following:

10. Who is George Boole?

a 19th century mathematician that developed boolean conditions

11. Precise set of rules for how to solve a problem is called a algorithm.

12. What are five things algorithms must have?

- have to in order
-have clear instructions
- produce a result
- stop in a finite amount of time
- operations that can be done more efficiently by the computer

13. Why do we analyze algorithms?

predict performance and help choose which algorithm is best to use