Academic Resource Sheet

CSIS 2101 Resources - Python

Basic Python "main" function:

```
Python

def main():
    print("Hello World!")

if __name__ == "__main__":
    main()
```

```
>>> a = 10.1234
>>> f'{a:.2f}'
'10.12'
```

Float Formatting using F-Format:

Using the Python Interpreter terminal, decimal numbers and floats can be formatted to fixed numbers using f-formatting.

This can also be printed directly using: print(f'{a:.2f}')

For-Loops in Python:

- Using Range():

Using Range(x), a For loop can repeat the statements within (x-1) number of times. For Example:

```
for x in range(6):

print(x)

This loop would print:

1
2
3
4
```

In general, the range function is used as follows: Range(a,b,c), with "a" being the beginning number of the counter, "b" being the number of incrementation after each iteration, and "c" being the limit counter number that shall not be reached. In the example above, "a" and "b" are omitted, defaulting to 1, but "c" is 6, effectively making the count stop at 5.

Calling functions in Python:

When a function is defined in a .py file, it can be called in the following way:

```
def my_function():
    print("Hello from a function")
```

Additionally, the function can also be called from another function's definition.

my_function()

<u>If-Statements in Python:</u>

```
a = 33
                                The "elif" keyword is
b = 33
                                python's way of
if b > a:
                                saying "if the
                                previous conditions
  print("b is greater than a")
                                were not true, then
elif a == b:
                                try this condition".
  print("a and b are equal")
b = 33
                                 The else keyword
if b > a:
                                 catches anything
  print("b is greater than a")
                                 which isn't caught
elif a == b:
                                 by the preceding
  print("a and b are equal")
                                 conditions.
  print("a is greater than b")
```

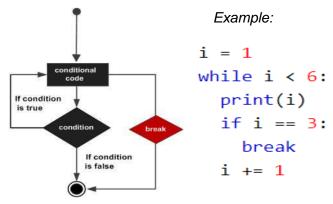
While-Loops in Python:

```
i = 1
while i < 6:
    print(i)
    i += 1</pre>
With the while-loop we can
execute a set of statements as
long as a condition is true.
```

In the example above, the while-loop block will keep on repeating over and over until the condition of i < 6 is broken. In other words, when i becomes a number bigger than of equal to 6.

Break Statements in Python:

A Break statement terminates the current loop and resumes execution at the next statement.



In the example above, the when the value of "i" becomes 3, the break statement will execute, and the loop would halt.

Contact us via:

