Quiz Prep Notes

Branching - Loops - Functions

Branching Class Notes

Boolean Expressions

|  |  |  |  |
| --- | --- | --- | --- |
| **4 == 2 \* 2** | **True** | **3<4 and 3>4** | **False** |
| **3 <= 4** | **True** | **3 < 4 or 3 > 4** | **True** |
| **3 != 4** | **True** | **3<4 or not (3>4)** | **True** |
| **12 % 2 == 0** | **True** |  |  |
| **11 % 2 ==0** | **False** |  |  |
| **not (3< 4 or not (3>4))** |  |  |  |
| **Not(3>4) = 3<=4** |  |  |  |
| **So… not( 3<4 or 3>4)** | **False** |  |  |
|  |  |  |  |

If/Else

|  |  |
| --- | --- |
| *# Code Snippet 2*  cond1 **=** **True**  **if** cond1:  print("Point A")    **else**:  print("Point B")    **print**("Point C")   * Point A * Point C | cond1 **=** **True**  cond2 **=** **True**  **if** cond1:  print("Point A")    **else**:  print("Point B")    **if** cond2:    print("Point C")    **print**("Point D")   * Point A * Point D |

if + elif

When an if is followed by one or more elif statements, the conditions following each of these are evaluated in order. As soon as one of the conditions is found to be true, the statements in the corresponding scope are executed. Then the statement following all of these is executed.



|  |  |
| --- | --- |
| cond1 **=** **True**  cond2 **=** **True**  cond3 **=** **True**  **if** cond1:  print("Point A")    **elif** cond2:  print("Point B")    **elif** cond3:  print("Point C")    **print**("Point D")   * Point A * Point D | cond1 = True  cond2 = True  cond3 = True  if cond1:  print("Point A")    if cond2:  print("Point B")    elif cond3:  print("Point C")    print("Point D")   * Point A * Point B  * Point D |

|  |  |
| --- | --- |
| cond1 **=** **True**  cond2 **=** **True**  cond3 **=** **True**  **if** cond1:  print("Point A")  **elif** cond2:  print("Point B")  **elif** cond3:  print("Point C")  **else**:  print("Point D")    print("Point E")   * Point A * Point E | cond1 **=** **False**  cond2 **=** **True**  cond3 **=** **True**  **if** cond1:  print("Point A")  **elif** cond2:  print("Point B")  **elif** cond3:  print("Point C")  **else**:  print("Point D")    print("Point E")   * Point B * Point E |

**for** i **in** range(5):

j **=** i**\*\***2

**if** i **==** 3:

**break**

print(i,j)

0 0

1 1

2 4

**for** i **in** range(5):

j **=** i**\*\***2

**if** i **==** 3:

**continue**

print(i,j)

0 0

1 1

2 4

Skips 3rd iteration

4 16

n\_list = [1,2,3,4]

sum = 0

for i in n\_list:

    sum += i

    mean = sum/len(n\_list)

print(mean)

n\_list **=** [1,**-**5,16,76,127,4]

lrgv **=** n\_list[0]

**for** i **in** n\_list:

**if** i **>** lrgv:

lrgv **=** i

print(lrgv)

**for** item **in** enumerate(alist):

print(item, type(item))

alist **=** ["Tom", "Dick", "Harry"]

alist = ["Tom", "Dick", "Harry"]

for item in enumerate(alist):

    value, index = item

    print(item, index, value)



Loops



Important: Notice when the loop is incremented--before or after it is evaluated or before or after it is printed.



|  |  |
| --- | --- |
| i **=** 0  **while** i **<** 5:  i **=** i **+** 1  print(i)   * 1 2 3 4 5   This type of problem is evaluated, incremented, and printed and then looped.  Not printed until it is incremented. |  The loop variable i could be initialized at either the first value we want or one before that value.   The < could have been <=   The change in the loop variabe could have been the last statement in the loop instead of the first. |
| i **=** 1  **while** i **<=** 5:  print(i)  i **=** i **+** 1   * 1 2 3 4 5 | i = 1  while i < 5:  print(i)  i = i + 1   * 1 2 3 4 |
|  |  |