



Conditions, Loops and Functions

An Introduction to Conditions, Loops and Functions

Conditions

Similar to other visual programming languages such as Scratch, the conditional statements that are used in python are: if, else, elif with comparison operators such as ==, !=, >, <, >=, <= .

IF statements will only run if the condition they are testing for is true.

```
age = 30
if age > 17 :
    print('You are old enough to drive')
```

Notice how the action that will be carried out if the condition is met is indented. This is how Python determines what is contained within the IF statement.

The ELSE works in a very similar way.

```
if age > 17 :
    print('You are old enough to drive')
else :
    print('You are not old enough to drive')
```

The ELIF is used to check for multiple conditions:

```
if age >= 21 :
    print('You are old enough to drive a tractor')
elif age >= 17:
    print('You are old enough to drive a car')
else :
    print('You are not old enough to drive')
```

Loops

Loops allows you to perform the same task multiple times.

FOR Loops

You can use loops to cycle through a list:

```
shopping_list = ['Juice', 'Tomatoes', 'Potatoes', 'Bananas']
for y in shopping_list:
    print(y)
```

You can also cycle through a list of values using a range:

```
for x in [2,4,6,8,10]:  
    print(x)
```

WHILE Loops

The While loop is useful in situations where you don't know the number of times you need to repeat the task.

```
random_number = random.randrange(0,100)  
  
while (random_number != 15):  
    print(random_number)  
    random_number = random.randrange(0,100)
```

Functions

Functions allow you to write and reuse the same piece of code. To set up a function you will need to type def, followed by the function's name, followed by the parameters it is to receive.

```
def addNumbers(fNum, sNum):  
    sumNum = fNum + sNum  
    return sumNum
```

```
print(addNumbers(1, 4))
```

You can also include any variables that have been declared outside of the function:

```
newNum = 0;  
def subNumbers(fNum, sNum):  
    newNum = fNum - sNum  
    return newNum
```

```
print(subNumbers(1, 4))
```