Python: part-2

Seham Eldeen

Comparison Operators | Logical Operators

==	1 == 2
!=	1 != 2
>	1 > 2
<	1 < 2
>=	1 >= 2
<=	1 <=2

and	1 == 2
or	1 != 2
not	1 > 2

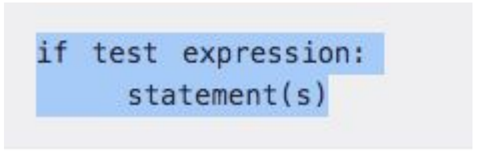
Decision Making

Decision making is done when we want to execute a certain part of our code only if a specific condition is satisfied.

1. If Statement
2. If… else Statements
3. If… elif… else statements
4. Nested If statement

If Statement

<u>Syntax:</u>



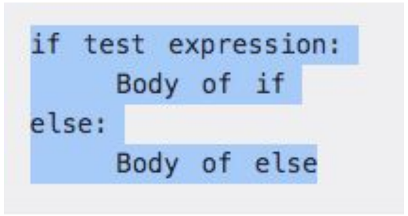
```
if test expression:
    statement(s)
```

<u>Note:</u>

- The above test expression is the condition
- Don’t forget to put the colon **:**
- Indentation is really important in python

If... else Statement

Syntax:



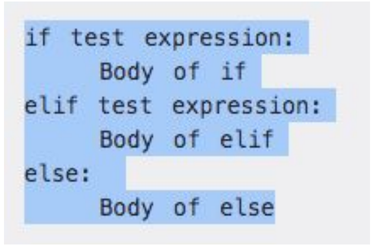
```
if test expression:
    Body of if
else:
    Body of else
```

Note:

- The above test expression is the condition
- Here, if the if condition is True, then the “Body of if” is executed, and if the condition is False, then “Body of else” is executed

If... elif... else Statement

Syntax:



```
if test expression:
     Body of if
elif test expression:
     Body of elif
else:
     Body of else
```

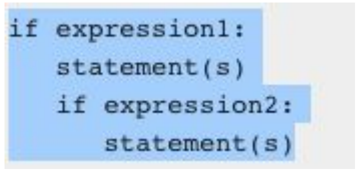
- elif is short for "else if". This "elif" allows us to create as many conditions as we want.

How this works?

- First the if condition is checked, and if it's False, it checks the condition of elif; if it's True, then "body of elif" is executed, and if it's False, then it jumps to the next elif condition and so on, and if all the conditions are False, then only the "body of else" is executed.

Nested If Statement

Syntax:



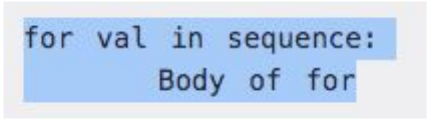
```
if expression1:
   statement(s)
   if expression2:
      statement(s)
```

Having an If statement or If… else statement or If…. elif… else statement inside another if statement

For loop

The for loop in is used to iterate over a **sequence** (list, dictionary, tuple, or any iterable object) or other iterable objects.

Syntax:



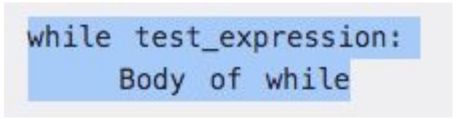
```
for val in sequence:
        Body of for
```

- Above, **val** is the variable that takes the value of the item inside the sequence on each iteration.
- The loop will continue to repeat till the last item in our sequence.

while loop

- allow us to continually perform an action as long as the condition is true.

Syntax:



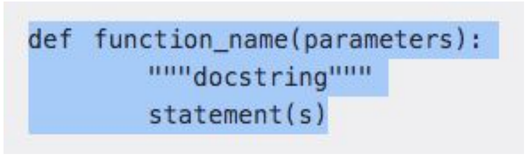
```
while test_expression:
    Body of while
```

- First, the condition is checked, and if the condition is True, then the body of while is evaluated. After that, the condition is checked again, and again, and this process only ends if the condition becomes False

Functions

- A function is a useful device that groups together a set of statements so they can be run more than once.us to continually perform an action as long as the condition is true.
- Functions allow us to write code once, and call several times

Syntax:



```
def  function_name(parameters):
        """docstring"""
        statement(s)
```

- “”” ……. “”” you can explain what the function does here.
- We can add parameters that can act as input in our function
- A docstring is short for documentation string, and is used to explain what a function does.

In Operator:

Returns True if a sequence with the specified value is present in the object

<u>Example:</u>

'x' **in** [1,2,3] ---> False
'x' **in** ['x','y','z'] ---> True

Exercises:

1. Write a function that will add two numbers together, only if they are even
2. Build a python calculator