

# Python Crash P2

## ▪ Constructs

- Comparison Operators
- If, elif, and else Statements
- For Loops
- While Loops
- range()
- List Comprehension
- Functions
- Lambda Expressions
- Map and Filter

## ▼ Comparison Operator

- Compare two elements >, <, ==, !=

```
In [47]: 1 == 2
```

```
Out[47]: False
```

```
In [49]: 1 != 3
```

```
Out[49]: True
```

```
In [50]: 'hi' == 'bye'
```

```
Out[50]: False
```

- Combine Logical Operators - and, or

```
In [55]: (1 < 2) and (2 > 3)
```

```
Out[55]: False
```

```
In [56]: (1 < 2) or (2 > 3)
```

```
Out[56]: True
```

## ▼ If, Elif and Else

- IF Statement

```
: if 1<2:  
    print('yep!')
```

```
yep!
```

- If else Statement

```
if 1 == 2:  
    print('First')  
else:  
    print('last')
```

- Multiple Condition If-Elif-Else Note: Execute first Elif which is true

```
if 1 == 2:  
    print('First')  
elif 4==4:  
    print('second')  
elif 3 == 3:  
    print('Middle')  
else:  
    print('last')
```

second

## ▼ For & While Loops

- For Loop

```
In [69]: seq = [1,2,3,4,5]
```

```
In [70]: for item in seq:  
         print(item)
```

1  
2  
3  
4  
5

- While Loop

```
i = 1  
  
while i < 5:  
    print('i is: {}'.format(i))  
    i = i+1
```

i is: 1  
i is: 2  
i is: 3  
i is: 4

## ▼ Useful functions

- Range

```
for x in range(0,5):  
    print(x)
```

0  
1  
2  
3  
4

```
list(range(10))
```

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

## ▼ List Comprehension

```
x = [1,2,3,4]
```

```
out = []
```

```
for num in x:  
    out.append(num**2)
```

```
print(out)
```

```
[1, 4, 9, 16]
```

- Better Approach with List Comprehension

```
[num**2 for num in x]
```

```
[1, 4, 9, 16]
```

## ▼ Functions

- keyword for defining function is def

```
def my_func(name='Default Name'):  
    print('Hello '+name)
```

```
my_func('Jose')
```

```
Hello Jose
```

- Function return value

```
def square(num):  
    return num**2
```

```
output = square(2)
```

```
output
```

```
4
```

- Doc String - SHIFT + TAB

```
def square(num):  
    """  
    THIS IS A DOCSTRING.  
    CAN GO MULTIPLE LINES.  
    THIS FUNCTION SQUARES A NUMBER.  
    """  
    return num**2
```

```
output = square(2)
```

```
output
```

```
4
```

```
square
```

Signature: square(num)

Docstring:

THIS IS A DOCSTRING.

CAN GO MULTIPLE LINES