

Variables

string : e.g. variablename = "john"
string : e.g. variablename = 'john'
integer: e.g. variablename = 0
float: e.g. variablename = 0.0
boolean: e.g. variablename = True
boolean: e.g. variablename = False

Print

```
print ("hello world")  
print ("hello world", myvariable)  
print ("hello world", myvariable, end = "")
```

Console input

```
myvariable = input("enter your name")  
myvariable = int (input("enter your age? "))  
myvariable = float( input("what is 3 / 4? "))
```

Decisions

```
#else and elif are optional  
if condition:  
    #your code here  
elif:  
    #your code here  
else:  
    #your code here
```

Conditional Operators

```
> >= < <= == != not , in, and  
e.g.:  
a != b  
not ( a==b)  
a == 1 or a == b or a == c  
a in (1,2,3)
```

Arithmetic operators

```
() * - + / % (modulus) ** ( exponential)
```

Assignment and Increments

```
= += *= -= /=
```

Lambda funtion

```
myvar = lambda arguments : expression  
x = lambda a : a + 10  
print(x(5))
```

Loops

```
while condition:  
    #your code here  
  
for in in range ( startvalue, stopvalue, increment):  
    #your code here
```

.Csv or .txt Files

```
myfile = open("filepath//filename.ext")  
for line in myfile:  
    print (line)  
    #your code here  
myfile.close()
```

Json Files

```
Write to a .json file  
with open('path//file.json', 'w') as outfile:  
    json.dump(mydict, outfile)  
Read from a .json file  
with open('path//filename.json') as infile:  
    mydict= json.load(json_data)
```

List and dictionaries

*python does not have built-in support for arrays,
but **python lists** can be used instead*

```
list, e.g.: mylist = [1,2,3]  
for i in range(0,le(mylist), 1):  
    print (i,mylist[i])  
for value in mylist:  
    print(value)
```

```
dictionary (key -value pairs), e.g.:  
mydict = {"name": "john", "age": 36 }  
mydict.keys()  
for key,val in mydict.items():  
    print (key,value)
```

Exception Handling

```
try:  
    #your code here  
except:  
    #your code here  
else:  
    #your code here  
finally:  
    #your code here
```