

In [1]: `print("hello world")`

hello world

In [2]: `Print("hello world")` *#python is case-sensitive*

```
-----
NameError                                Traceback (most recent call last)
Cell In[2], line 1
----> 1 Print("hello world")

NameError: name 'Print' is not defined
```

In [3]: `print(hello world)` *#it should be in inverted comma for string*

```
Cell In[3], line 1
    print(hello world) #it should be in inverted comma
      ^
SyntaxError: invalid syntax. Perhaps you forgot a comma?
```

In [4]: `print(10)`

10

In [5]: `print(10.7)`

10.7

`;agvvshvhvsvhsg`

In [6]: *###;agvvshvhvsvhsg #keep the cursor on in[] and press m and run that*

```
-----
NameError                                Traceback (most recent call last)
Cell In[6], line 1
----> 1 agvvshvhvsvhsg("#keep the cursor on in[] and press m and run that")

NameError: name 'agvvshvhvsvhsg' is not defined
```

`sgssvghvshh`

`printhello`

`sbhsb`

`shbshh`

`shhsjhj`

In [10]: `#sjsbj`

`jhsjsjs`

`bjsbjbsj`

`ahhsj`

In [11]: *### 1.variable name should not have space*

```
First name= "alok singh"

Cell In[11], line 2
    First name= "alok singh"
      ^
SyntaxError: invalid syntax
```

In [13]: *### 2.variable name should start with A-Z ,a-z or \_*

```
First name= "alok singh"

Cell In[13], line 2
    First name= "alok singh"
      ^
SyntaxError: invalid syntax
```

In [14]: `First name= "alok singh"`

```
Cell In[14], line 1
    First name= "alok singh"
      ^
SyntaxError: invalid syntax
```

In [15]: `First name="alok singh"`

```
Cell In[15], line 1
    First name="alok singh"
      ^
SyntaxError: invalid syntax
```

In [16]: `First name="aloksingh"`

```
Cell In[16], line 1
    First name="aloksingh"
      ^
SyntaxError: invalid syntax
```

In [17]: `Firstname="alok singh"`  
`name="alok"`  
`_name="alok"`

In [18]: `Firstname="alok singh"`  
`name="alok"`  
`_name="alok"`

In [19]: *#variable name shoud not start with number but number cam be present anywhere*

In [20]: *#and only allowed speacial charater is \_ in variable*

## Data types

In [21]: `x=10.5`

`type(x)`

Out[21]: `float`

In [22]: `y=10`

`type(y)`

Out[22]: `int`

In [23]: `z=True` *#use capital T*

`type(z)`

Out[23]: `bool`

In [24]: `x=10` *#it will be executing only last line*

`type(x)`

`x`

Out[24]: `10`

In [25]: `x,type(x)`

Out[25]: `(10, int)`

In [26]: `x,y=10,20`

`x`

`y`

Out[26]: `20`

In [27]: `x,y`

Out[27]: `(10, 20)`

In [28]: `y,x`

Out[28]: `(20, 10)`

## Data type conversion

(1) implicit-> python interpreter without programmer's intervention

(2) explicit ->user-defined ->casting

In [29]: *#implicit conversion*

*# int and float*

`x=1.5`

`y=3`

`z=x*y`

`type(z)`

Out[29]: `float`

In [30]: `x="data"`

`y=1.5`

`z=x*y`

`type(z)`

```
-----
TypeError                                Traceback (most recent call last)
Cell In[30], line 3
      1 x="data"
      2 y=1.5
----> 3 z=x*y
      4 type(z)

TypeError: can't multiply sequence by non-int of type 'float'
```

In [31]: `x=20`

`y=10`

`z=x/y`

`type(z)`

Out[31]: `float`

In [32]: `z`

Out[32]: `2.0`

In [33]: *#explicit conversion*

`x="2"`

`y=1.5`

`z=x+y`

`type(z)`

```
-----
TypeError                                Traceback (most recent call last)
Cell In[33], line 4
      2 x="2"
      3 y=1.5
----> 4 z=x+y
      5 type(z)

TypeError: can only concatenate str (not "float") to str
```

In [34]: `z=int(x)+y`

`z`

Out[34]: `3.5`

## type casting

`int(),float(),bool()`

`,str()`

In [35]: `int-> float ->possible`

`int->bool ->possible`

`int ->string->possible`

```
Cell In[35], line 1
    int-> float ->possible
      ^
SyntaxError: invalid syntax
```

In [36]: `x=5`

`str(x)`

Out[36]: `'5'`

In [37]: `x=0`

`bool(x)`

Out[37]: `False`

In [ ]: