

- code to markdown format -- esc+m
- markdown to code format -- esc+y
- execute the cell -- shift+enter

Heading1

Heading2

Heading3

Heading6

Displaying an image

- syntax : ![image name](image link)



Link

[Apssdc portal link \(http://engineering.apssdc.in/\)](http://engineering.apssdc.in/)

Creating a variable

```
In [1]: a = 10
```

```
In [2]: 1a = 10
```

File "<ipython-input-2-1be9f8edb7cd>", line 1

```
1a = 10
^
```

SyntaxError: invalid syntax

```
In [3]: name = 23
```

Data Types:

- integer
- float
- string
- Boolean

```
In [4]: a = 10  
type(a)
```

Out[4]: int

```
In [6]: b = 10.7  
print(type(b))
```

<class 'float'>

```
In [7]: c = "apssdc@123"  
type(c)
```

Out[7]: str

```
In [8]: d = 'y'  
type(d)
```

Out[8]: str

```
In [11]: e = True  
type(e)
```

Out[11]: bool

Conversions of datatypes

```
In [13]: type(a)
```

Out[13]: int

```
In [15]: print(float(a))  
print(type(float(a)))
```

10.0
<class 'float'>

```
In [16]: b
```

Out[16]: 10.7

```
In [17]: int(b)
```

```
Out[17]: 10
```

```
In [19]: f = "1234a"  
int(f)
```

```
-----  
ValueError                                Traceback (most recent call last)  
<ipython-input-19-1b89b2abf4ef> in <module>  
      1 f = "1234a"  
----> 2 int(f)  
  
ValueError: invalid literal for int() with base 10: '1234a'
```

```
In [22]: a,b,c = 10,20,40  
a
```

```
Out[22]: 10
```

```
In [23]: a = 10
```

```
In [26]: input()
```

```
Out[26]: ''
```

```
In [27]: s = input("enter your name")
```

```
enter your nameapssdc
```

```
In [28]: type(s)
```

```
Out[28]: str
```

```
In [29]: input("enter a value")
```

```
enter a value56
```

```
Out[29]: '56'
```

```
In [31]: x = int(input("enter a value"))  
x
```

```
enter a value56
```

```
Out[31]: 56
```

```
In [32]: type(x)
```

```
Out[32]: int
```

Operators

- Operators is used to perform different operations on the variables or values
- Types of Operators:
 - Arithmetic operators : +, -, *, /, %, //
 - Assignment operators : =, +=, -=, %=, /=, //=
 - comparison operators : ==, >=, > !=, >, <
 - Logical operators : And, or, Not
 - Bitwise operators : left shift, right shift
 - Identity operators : is, is not
 - Membership operators : in, not in

```
In [34]: a = 10  
b = 11  
b += a # b=b+a  
b
```

```
Out[34]: 21
```

```
In [35]: b -= a # b=b-a  
b
```

```
Out[35]: 11
```

- Identity operators

```
In [36]: a is b
```

```
Out[36]: False
```

```
In [42]: a == b
```

```
Out[42]: False
```

```
In [43]: a != b
```

```
Out[43]: True
```

```
In [37]: a is not b
```

```
Out[37]: True
```

- Membership operator

```
In [38]: a = "apssdc"
         b = "1"
         b in a
```

Out[38]: False

```
In [39]: '1' in a
```

Out[39]: False

```
In [40]: '1' in "apssdc"
```

Out[40]: False

```
In [41]: '1' not in "apssdc"
```

Out[41]: True

Conditional Programming

- if
- elif
- else

```
In [47]: # write a program which number is greater
```

```
a = int(input("enter a value"))
b = int(input("enter b value"))
if a>b:
    print(a,"is greater value")
else:
    print(b,"is greater value")
```

```
enter a value89
enter b value90
90 is greater value
```

```
In [51]: # find the greatest value of three numbers
```

```
a = 10
b = 20
c = 30
if a>b and a<c:
    print(a)
elif b>a and b>c:
    print(b)
else:
    print(c)
```

30

Loops

- for
 - for variable in iterator
- while
 - while condition

```
In [57]: a = 10

for i in range(1,a+1): # start value,end value,step value
    print(i,end = " ")

1 2 3 4 5 6 7 8 9 10
```

```
In [59]: a = 10

for i in range(1,a+1,5): # start value,end value,step value
    print(i,end = " ")

1 6
```

```
In [ ]: 1
        1+5=6
        6+5=11
```

```
In [67]: for i in range(a,0,-1):
        print(i,end = " ")

10 9 8 7 6 5 4 3 2 1
```

```
In [68]: for i in range(10,0,-3):
        print(i,end = " ")

10 7 4 1
```

```
In [69]: a = "apssdc"
        len(a)
```

Out[69]: 6

```
In [71]: a = "apssdc"
        for i in range(len(a)):
            print(a[i],end = " ")

a p s s d c
```

```
In [73]: for i in a:
        print(i,end="")

apssdc
```

```
In [74]: a = "string"
b = "899876"
for i in a:
    for j in b:
        print(i,j)
```

...

```
In [75]: a = 1
while a<10:
    print(a)
    a+=1#a=a+1
```

1
2
3
4
5
6
7
8
9

```
In [76]: a = 1
while a<10:
    print(a)
    if a==5:
        break
    a+=1
```

1
2
3
4
5

```
In [77]: a = 1
while a<10:
    print(a)
    if a==5:
        continue
    a+=1
```

...

Data Structures:

- List
- Tuples
- Dictionary
- String
- Set

Strings

- string is immutable

```
In [78]: a = "string$1345"  
         type(a)
```

```
Out[78]: str
```

```
In [79]: len(a)
```

```
Out[79]: 11
```

- Indexing in strings
 - forward indexing --> 0 to len(string)-1
 - Backward indexing --> -len(string) to -1

```
In [80]: a
```

```
Out[80]: 'string$1345'
```

```
In [83]: len(a)
```

```
Out[83]: 11
```

```
In [81]: a[0]
```

```
Out[81]: 's'
```

```
In [82]: a[90]
```

```
-----  
IndexError                                Traceback (most recent call last)  
<ipython-input-82-1782cd4009e4> in <module>  
----> 1 a[90]  
  
IndexError: string index out of range
```

```
In [84]: a[-1]
```

```
Out[84]: '5'
```

```
In [85]: a
```

```
Out[85]: 'string$1345'
```



```
In [86]: a[-4]
```

```
Out[86]: '1'
```

```
In [87]: a[5]
```

```
Out[87]: 'g'
```

slicing

```
In [91]: a[0:4]
```

```
Out[91]: 'stri'
```

```
In [89]: a
```

```
Out[89]: 'string$1345'
```

```
In [92]: a[2:6]
```

```
Out[92]: 'ring'
```

```
In [93]: a[-3:]
```

```
Out[93]: '345'
```

```
In [94]: a[0:10:2]
```

```
Out[94]: 'srn$3'
```

```
In [ ]: a = "welcome to workshop"
```