

python type casting

```
In [1]: int(2.3)
```

```
Out[1]: 2
```

```
In [3]: int(True)
```

```
Out[3]: 1
```

```
In [5]: int(False)
```

```
Out[5]: 0
```

```
In [7]: int('10')
```

```
Out[7]: 10
```

float

```
In [8]: float(10)
```

```
Out[8]: 10.0
```

```
In [10]: float(True)
```

```
Out[10]: 1.0
```

```
In [11]: float(False)
```

```
Out[11]: 0.0
```

```
In [13]: float(10)
```

```
Out[13]: 10.0
```

complex

```
In [15]: complex(10)
```

```
Out[15]: (10+0j)
```

```
In [1]: complex(10,20)
```

```
Out[1]: (10+20j)
```

```
In [2]: complex(2.3)
```

```
Out[2]: (2.3+0j)
```

```
In [3]: complex(2.3,3.4)
```

```
Out[3]: (2.3+3.4j)
```

```
In [6]: complex('10')
```

```
Out[6]: (10+0j)
```

```
In [8]: complex(10,int('10'))
```

```
Out[8]: (10+10j)
```

```
In [9]: complex(True,False)
```

```
Out[9]: (1+0j)
```

```
In [10]: complex(False,True)
```

```
Out[10]: 1j
```

```
In [11]: complex(False,False)
```

```
Out[11]: 0j
```

bool

```
In [12]: bool(1)
```

```
Out[12]: True
```

```
In [13]: bool()
```

```
Out[13]: False
```

```
bool(1+2j)
```

```
In [17]: print(str(1))  
print(str(1.2))  
print(str(True))  
print(str(1+2j))
```

```
1  
1.2  
True  
(1+2j)
```

string indexing

_forward indexing

_backward indexing

_step indexing

```
In [19]: s = 'hello'
```

```
In [20]: s
```

```
Out[20]: 'hello'
```

```
In [21]: s[-1]
```

```
Out[21]: 'o'
```

```
In [22]: s
```

```
Out[22]: 'hello'
```

```
In [26]: s[4]
```

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

```
In [25]: s
```

```
Out[25]: 'hello'
```

```
In [27]: s[:]
```

```
Out[27]: 'hello'
```

```
In [29]: s[0:1]
```

```
Out[29]: 'h'
```

```
In [30]: s[0:2]
```

```
Out[30]: 'he'
```

```
In [31]: s
```

```
Out[31]: 'hello'
```

```
In [32]: print(s[0])  
print(s[1])  
print(s[2])  
print(s[3])  
print(s[4])
```

```
h  
e  
l  
l  
o
```

```
In [33]: s
```

```
Out[33]: 'hello'
```

```
In [34]: for i in s:  
         print(i)
```

```
h  
e  
l  
l  
o
```

```
In [41]: s
```

```
Out[41]: 'hello'
```

```
In [42]: s1 = 'nareshit'
```

```
In [43]: s + s1
```

```
Out[43]: 'hellonareshit'
```

```
In [45]: s3 = s + s1  
s3
```

```
Out[45]: 'hellonareshit'
```

```
In [46]: s
```

```
Out[46]: 'hello'
```

```
In [47]: s[2:5]
```

```
Out[47]: 'llo'
```

```
In [48]: s2 = 'r','g','y'  
s2
```

```
Out[48]: ('r', 'g', 'y')
```

```
In [50]: s2[1:4]
```

```
Out[50]: ('g', 'y')
```

```
In [ ]:
```