

Survival functions

- type
- help
- dir

what is string?

- String is a collection of characters

In [1]:

```
# how do we define single line strings?
```

```
'prashant'  
"prashant"
```

```
# how do we define multi-line strings?
```

```
'''prashant  
velocity'''
```

```
"""prashant  
velocity"""
```

Out[1]:

```
'prashant\nvelocity'
```

In []:

```
foo = "prashant python velocity"
```

Everything in Python is a object

Examples of object -

Object vs Class

apple_smartwatch	=>	gadgets
samsung_mobile	=>	Mobile
Modi	=>	Human
Mark Zuckerberg	=>	Engineer
World_cup	=>	Tournament

In [2]:

```
foo = "prashant python velocity"  
type(foo)  # `foo` is a object of class `str`
```

Out[2]:

str

In [4]:

```
bar = 100  
type(bar)  # `bar` is a object of class `int`
```

Out[4]:

int

In [5]:

```
example1 = 100.50  
type(example1)  # `example1` is an object of class `float`
```

Out[5]:

float

In [7]:

```
example2 = True  
type(example2)  # `example2` is an object of class `bool`
```

Out[7]:

bool

In [10]:

```
foo = "prashant python velocity"  
type(foo)  # `foo` is a object of class `str`  
print(foo)
```

prashant python velocity

`dir()` function

- will return attributes associated with object

Terminology

attribute -

- function associated with object
- data associated with object
- attribute is anything that can be accessed using . DOT operator

In [11]:

```
dir(foo) # attributes associated with object `foo`  
        # `foo` is an object of class `str`
```

Out[11]:

```
['__add__',  
 '__class__',  
 '__contains__',  
 '__delattr__',  
 '__dir__',  
 '__doc__',  
 '__eq__',  
 '__format__',  
 '__ge__',  
 '__getattr__',  
 '__getitem__',  
 '__getnewargs__',  
 '__gt__',  
 '__hash__',  
 '__init__',  
 '__init_subclass__',  
 '__iter__',  
 '__le__',  
 '__len__',  
 '__lt__',  
 '__mod__',  
 '__mul__',  
 '__ne__',  
 '__new__',  
 '__reduce__',  
 '__reduce_ex__',  
 '__repr__',  
 '__rmod__',  
 '__rmul__',  
 '__setattr__',  
 '__sizeof__',  
 '__str__',  
 '__subclasshook__',  
 'capitalize',  
 'casefold',  
 'center',  
 'count',  
 'encode',  
 'endswith',  
 'expandtabs',  
 'find',  
 'format',  
 'format_map',  
 'index',  
 'isalnum',  
 'isalpha',  
 'isascii',  
 'isdecimal',  
 'isdigit',  
 'isidentifier',  
 'islower',  
 'isnumeric',  
 'isprintable',  
 'isspace',
```

```
'istitle',  
'isupper',  
'join',  
'ljust',  
'lower',  
'lstrip',  
'maketrans',  
'partition',  
'removeprefix',  
'removesuffix',  
'replace',  
'rfind',  
'rindex',  
'rjust',  
'rpartition',  
'rsplit',  
'rstrip',  
'split',  
'splitlines',  
'startswith',  
'strip',  
'swapcase',  
'title',  
'translate',  
'upper',  
'zfill']
```

In [13]:

```
# how to create a empty string?
```

```
random = ''  
type(random) # `random` is an object of class `str`  
dir(random)
```

Out[13]:

```
['_add__',  
 '__class__',  
 '__contains__',  
 '__delattr__',  
 '__dir__',  
 '__doc__',  
 '__eq__',  
 '__format__',  
 '__ge__',  
 '__getattribute__',  
 '__getitem__',  
 '__getnewargs__',  
 '__gt__',  
 '__hash__',  
 '__init__',  
 '__init_subclass__',  
 '__iter__',  
 '__le__',  
 '__len__',  
 '__lt__',  
 '__mod__',  
 '__mul__',  
 '__ne__',  
 '__new__',  
 '__reduce__',  
 '__reduce_ex__',  
 '__repr__',  
 '__rmod__',  
 '__rmul__',  
 '__setattr__',  
 '__sizeof__',  
 '__str__',  
 '__subclasshook__',  
 'capitalize',  
 'casefold',  
 'center',  
 'count',  
 'encode',  
 'endswith',  
 'expandtabs',  
 'find',  
 'format',  
 'format_map',  
 'index',  
 'isalnum',  
 'isalpha',  
 'isascii',  
 'isdecimal',  
 'isdigit',  
 'isidentifier',  
 'islower',
```

```
'isnumeric',
'isprintable',
'isspace',
'istitle',
'isupper',
'join',
'ljust',
'lower',
'lstrip',
'maketrans',
'partition',
'removeprefix',
'removesuffix',
'replace',
'rfind',
'rindex',
'rjust',
'rpartition',
'rsplit',
'rstrip',
'split',
'splitlines',
'startswith',
'strip',
'swapcase',
'title',
'translate',
'upper',
'zfill']
```

functions associated with `string` object

'capitalize', 'casefold', 'center', 'count', 'encode', 'endswith', 'expandtabs', 'find', 'format', 'format_map', 'index', 'isalnum', 'isalpha', 'isascii', 'isdecimal', 'isdigit', 'isidentifier', 'islower', 'isnumeric', 'isprintable', 'isspace', 'istitle', 'isupper', 'join', 'ljust', 'lower', 'lstrip', 'maketrans', 'partition', 'removeprefix', 'removesuffix', 'replace', 'rfind', 'rindex', 'rjust', 'rpartition', 'rsplit', 'rstrip', 'split', 'splitlines', 'startswith', 'strip', 'swapcase', 'title', 'translate', 'upper', 'zfill'

how do we invoke `functions` associated with `object`?

syntax

```
<object-name>.<function-name>( )
```

In [16]:

```
name = "prashant"
type(name) # `name` is an object of class `str`
```

Out[16]:

str

In [21]:

```
print(name)  # original
result = name.upper()
print(name)  # after function call
result
```

```
prashant
prashant
```

Out[21]:

```
'PRASHANT'
```

In [20]:

```
result.lower()
```

Out[20]:

```
'prashant'
```

In []:

```
Human
- data
  - height
  - weight
- functions
  - eat
  - sleep
  - drink
  - work

object      class
virat  -----  Human
hardik -----  Human
```