Topics:

- strings
- functions

Strings:

- indexing
- string slicing
- string methods

In [1]:
```
name1 = 'Tyson'
name1
```

Out[1]: 'Tyson'

String indexing:

- positive indexing.
  - index start from zero.
    T|y|s|o|n
    0 1 2 3 4
- negative indexing.
    T |y |s |o |n
    -5 -4 -3 -2 -1

In [2]:
```
a = 'Dragoon '
```

In [3]:
```
a
```

Out[3]: 'Dragoon'

In [4]:
```
a[3]
```

Out[4]: 'g'

In [12]:
```
len(a)
```

Out[12]: 7

In [8]:
```
a[6]
```

Out[8]: 'n'

In [9]:
```
a[-1]
```

Out[9]: 'n'

In [10]:
```
a[-4]
```

Out[10]: 'g'

In [14]:
```
b = input('enter any one string: ')
print('the last character is: ',b[-1])
```

```
enter any one string: vrsec
the last character is:  c
```

String slicing:

synax:

string_obj[start:'included':end_ 'not included':range]

In [15]:
```
print(b)
```

```
vrsec
```

In [16]:
```
b[0:4:2] # 2 ===> is a range.
```

Out[16]: 'vs'

In [17]:
```
a[1:4]
```

Out[17]: 'rag'

In [18]:
```
a[0:]
```

Out[18]: 'Dragoon'

In [19]:
```
a[0::2]
```

Out[19]: 'Daon'

In [20]:
```
a[1:6:3]
```

Out[20]: 'ro'

In [21]:
```
a[-3:-1]
```

Out[21]: 'oo'

In [22]:
```
a[-3:0]
```

Out[22]: ''

In [23]:
```
a[-3:-1:-1]
```

Out[23]: ''

In [25]:
```
a[-4:-1]
```

Out[25]: 'goo'

In [26]:
```
a[:-1]
```

Out[26]: 'Dragoo'

In [27]:
```
a[:0]
```

Out[27]: ''

In [28]:
```
a[-6:-1]
```

Out[28]: 'ragoo'

In [29]:
```
a[:-2]
```

Out[29]: 'Drago'

In [30]:
```
a[0:]
```

Out[30]: 'Dragoon'

In [31]:
```
a[-3:]
```

Out[31]: 'oon'

Arithimatic operations on strings

In [32]:
```
a
```

Out[32]: 'Dragoon'

In [33]:
```
name1
```

Out[33]: 'Tyson'

In [34]:
```
a + name1
```

Out[34]: 'DragoonTyson'

In [36]:
```
a + ' ' + name1
```

Out[36]: 'Dragoon Tyson'

In [37]:
```
a + ' strom'
```

Out[37]: 'Dragoon strom'

In [39]:
```
name1 + " Gramger"
```

Out[39]: 'Tyson Gramger'

inbuilt functions on strings

In [40]:
```
len(a) # to calculate length of the string
```

Out[40]: 7

In [42]:
```
result = " ".join(name1)
print(result)
```

```
T@y@s@o@n
```

In [43]:
```
answer = "@ ".join(a)
print(answer)
```

```
D@ r@ a@ g@ o@ o@ n
```

In [44]:
```
min(a) # based on ascii value of the character.
```

Out[44]: 'D'

In [46]:
```
ord('a')
```

Out[46]: 97

In [47]:
```
ord('A')
```

Out[47]: 65

In [48]:
```
max(name1)
```

Out[48]: 'y'

Sting methods

In [50]:
```
s = 50
dir(s)
```

Out[50]:
```
['__abs__',
 '__add__',
 '__and__',
 '__bool__',
 '__ceil__',
 '__class__',
 '__delattr__',
 '__dir__',
 '__divmod__',
 '__doc__',
 '__eq__',
 '__float__',
 '__floor__',
 '__floordiv__',
 '__format__',
 '__ge__',
 '__getattribute__',
 '__getnewargs__',
 '__gt__',
 '__hash__',
 '__index__',
 '__init__',
 '__init_subclass__',
 '__int__',
 '__invert__',
 '__le__',
 '__lshift__',
 '__lt__',
 '__mod__',
 '__mul__',
 '__ne__',
 '__neg__',
 '__new__',
 '__or__',
 '__pos__',
 '__pow__',
 '__radd__',
 '__rand__',
 '__rdivmod__',
 '__reduce__',
 '__reduce_ex__',
 '__repr__',
 '__rfloordiv__',
 '__rlshift__',
 '__rmod__',
 '__rmul__',
 '__ror__',
 '__round__',
 '__rpow__',
 '__rrshift__',
 '__rshift__',
 '__rsub__',
 '__rtruediv__',
```

```
'__rxor__',
'__setattr__',
'__sizeof__',
'__str__',
'__sub__',
'__subclasshook__',
'__truediv__',
'__trunc__',
'__xor__',
'as_integer_ratio',
'bit_length',
'conjugate',
'denominator',
'from_bytes',
'imag',
'numerator',
'real',
'to_bytes']
```

In [51]:
```
dir(name1)
```

Out[51]:
```
['__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',
 'replace',
 'rfind',
 'rindex',
 'rjust',
 'rpartition',
 'rsplit',
 'rstrip',
 'split',
 'splitlines',
 'startswith',
 'strip',
 'swapcase',
 'title',
 'translate',
 'upper',
 'zfill']
```

In [52]:
```
'aDfSEYUghj'.casefold()
```

Out[52]:
```
'adfseyughj'
```

In [53]:
```
'aDfSEYUghj'.swapcase()
```

Out[53]:
```
'AdFseyuGHJ'
```

In [54]:
```
'evil dragoon'.capitalize()
```

Out[54]:
```
'Evil dragoon'
```

In [55]:
```
a.title()
```

Out[55]:
```
'Dragoon'
```

In [56]:
```
a
```

Out[56]:
```
'Dragoon'
```

In [57]:
```
s1 = 'drigger'
s1.title()
```

Out[57]:
```
'Drigger'
```

In [59]:
```
help('translate')
```

```
No Python documentation found for 'translate'.
Use help() to get the interactive help utility.
Use help(str) for help on the str class.
```

In [62]:
```
ord('s'),ord('a'),ord('i')
```

Out[62]:
```
(115, 97, 105)
```

In [63]:
```
ord('S'), ord('A'), ord('I')
```

Out[63]:
```
(83, 65, 73)
```

In [64]:
```
s1.count('e')
```

Out[64]:
```
1
```

In [65]:
```
s1.count('g')
```

Out[65]:
```
2
```

In [66]:
```
help(''.translate)
```

```
Help on built-in function translate:

translate(table, /) method of builtins.str instance
    Replace each character in the string using the given translation table.

      table
        Translation table, which must be a mapping of Unicode ordinals to
        Unicode ordinals, strings, or None.

    The table must implement lookup/indexing via __getitem__, for instance a
    dictionary or list.  If this operation raises LookupError, the character is
    left untouched.  Characters mapped to None are deleted.
```

In [67]:
```
help(''.ljust)
```

```
Help on built-in function ljust:

ljust(width, fillchar=' ', /) method of builtins.str instance
    Return a left-justified string of length width.

    Padding is done using the specified fill character (default is a space).
```

In [71]:
```
name1.ljust(20,'$')
```

Out[71]:
```
'Tyson$$$$$$$$$$$$$$'
```

In [73]:
```
a.ljust(10,'!')
```

Out[73]:
```
'Dragoon!!!'
```

In [74]:
```
'12345678910'.isdigit() # is a purly digit
```

Out[74]:
```
True
```

In [75]:
```
'12345asdfrgdfhd'.isdigit() # is not a purly digit
```

Out[75]:
```
False
```

In [76]:
```
help(''.split)
```

```
Help on built-in function split:

split(sep=None, maxsplit=-1) method of builtins.str instance
    Return a list of the words in the string, using sep as the delimiter strin
g.

    sep
      The delimiter according which to split the string.
      None (the default value) means split according to any whitespace,
      and discard empty strings from the result.
    maxsplit
      Maximum number of splits to do.
      -1 (the default value) means no limit.
```

In [83]:
```
'maxy'.split()
```

Out[83]:
```
['maxy']
```

In [87]:
```
v = 'sai\t'
v.expandtabs(tabsize = 100)
```

Out[87]:
```
'sai
'
```

In [88]:
```
len(v)
```

Out[88]:
```
4
```

In [89]:
```
help(''.islower)
```

```
Help on built-in function islower:

islower() method of builtins.str instance
    Return True if the string is a lowercase string, False otherwise.

    A string is lowercase if all cased characters in the string are lowercase a
nd
    there is at least one cased character in the string.
```

In [90]:
```
'sai tyson ray'.islower()
```

Out[90]:
```
True
```

In [92]:
```
'Sai'.islower() # due to capital letter.
```

Out[92]:
```
False
```

In [94]:
```
'tyson,ray,max,kai'.split(',')
```

Out[94]:
```
['tyson', 'ray', 'max', 'kai']
```

In [95]:
```
'tyson,ray,max,kai'.split('a')
```

Out[95]:
```
['tyson,r', 'y,m', 'x,k', 'i']
```

Functions in python

to create a function we need to use 'def' keyword

In [96]:
```
help(def)
```

```
  File "<ipython-input-96-083659f89be1>", line 1
    help(def)
         ^
SyntaxError: invalid syntax
```

In [ ]:
```

```