

Programming and Problem Solving through Python Language O Level / A Level

Chapter - 6 : Functions

String Functions

A string is a sequence of characters enclosed in quotation marks.

22. isalnum() Function

This method returns True if all the characters are alphanumeric means alphabet[A-Z,a-z] or numeric[0-9], otherwise False.

Syntax `string.isalnum( )`

Example

```
obj = "Abc72"
x=obj.isalnum()
print(x)

obj = "Abc"
x=obj.isalnum()
print(x)

obj = "Abc 72"
x=obj.isalnum()
print(x)
```

Output

```
True
True
False
```

23. startswith() Function

This method returns True if all the string start with specified value, otherwise False.

Syntax `string.startswith(value, start, end )`

value(Required)	- A String	-Old string want to replace.
start(Optional)	- An Integer	-Start Position.
end(Optional)	- An Integer	-End Position.

Example

```
obj = "Apple a day , keeps Doctor Away"
x=obj.startswith("Apple")
print(x)

x=obj. startswith("Doctor",20)
print(x)
```

Output

```
True
True
```

24. endswith() Function

This method returns True if all the string ends with specified value, otherwise False.

Syntax string.endswith(value, start, end)

value(Required)	- A String	-Old string want to replace.
start(Optional)	- An Integer	-Start Position.
end(Optional)	- An Integer	-End Position.

Example

```
obj = "Apple a day , keeps Doctor Away"
x=obj.endswith("Away")
print(x)
x=obj.endswith("day",1,11)
print(x)
```

Output

```
True
True
```

25. encode() Function

This method converts the string with the specified encoding. The default encoding scheme is UTF-8(Unicode Transformation Format).

Syntax string.encode(encoding, errors)

encoding(Optional)	- A String	-Encoding Scheme(utf-8, ascii,utf-32 etc)
errors(Optional)	- A String	-When encoding fails(strict,ignore,replace).

Example

```
obj = "Apple a day! \@Doctor"
x=obj.encode()
print("utf-8 : ", x)

x=obj.encode(encoding="ascii")
print("ascii : ",x)

x=obj.encode(encoding="utf-32")
print("utf-32 : ",x)
```

Output

```
utf-8 : b'Apple a day! \@Doctor'
ascii : b'Apple a day! \@Doctor'
utf-32 :
b'\xff\xfe\x00\x00A\x00\x00\x00p\x00\x00\x00p\x00\x00\x00l\x00\x00\x00e\x00
\x00\x00 \x00\x00\x00a\x00\x00\x00
\x00\x00\x00d\x00\x00\x00a\x00\x00\x00y\x00\x00\x00!\x00\x00\x00
\x00\x00\x00\x00\x00@\x00\x00\x00D\x00\x00\x00o\x00\x00\x00c\x00\x00
0\x00t\x00\x00\x00o\x00\x00\x00r\x00\x00\x00'
```

26. decode() Function

This method converts the string with the specified encoding. The default encoding scheme is UTF-8(Unicode Transformation Format).

Syntax `string.decode(encoding, error )`

<code>encoding(Optional)</code>	- A String	-Encoding Scheme(utf-8, ascii,utf-32 etc)
<code>error(Optional)</code>	- A String	-Error Options when decoding fails.

Example

```
obj = "Apple a day! \@Doctor"
x=obj.encode( )
y=x.decode( )
print("utf-8 : ", y)

x=obj.encode(encoding="ascii")
y=x.decode(encoding="ascii" )
print("ascii : ",y)

x=obj.encode(encoding="utf-32")
y=x.decode(encoding="utf-32"))
print("utf-32 : ",y)
```

Output

```
utf-8 : Apple a day! \@Doctor
ascii : Apple a day! \@Doctor
utf-32 : Apple a day! \@Doctor
```

String Slicing

- Strings in are arrays of bytes representing unicode characters.
- We can extract a range of characters by using the slice syntax.
- To slice the parts of string specify the start index and the end index, separated by a colon.

Example

```
a = "Hello, Python!"
b=a[2:9]        # start : end
print(b)

b=a[2:9:2]    #start : end : increment
print(b)

b=a[-7:-1]    #negative indexing , -1 refer to last character
print(b)

b=a[-7:-1:2]
print(b)
```

Output

```
llo, Py
lo y
Python
Pto
```

String Membership

We can use the "in" or "not in" in Membership operator to check if a substring is present in a string

Example

```
a = "Hello, Python!"  
b="Hello" in a  
print(b)  
  
b="Wello" not in a  
print(b)
```

Output

```
True  
True
```