

Introduction to Python

Basic Syntax

Datatypes II: Strings, NoneType

Topics

- 1) Strings (indexing, slicing)
- 2) NoneType

String

Strings in Python are created with single or double quotes.

```
In [17]: message = 'what do you like?'  
         response = 'spam'
```

```
In [18]: len(response)
```

```
Out [18]: 4
```

```
In [19]: # Make uppercase. See also str.lower()  
         response.upper()
```

```
Out [19]: 'SPAM'
```

String Concatenation

```
In [21]: # concatenation with +  
         message + response
```

```
Out [21]: 'what do you like?spam'
```

```
In [22]: # multiplication is multiple concatenation  
         5 * response
```

```
Out [22]: 'spamspamspamspamspam'
```

String Indexing

```
In [23]: message = "what do you like?"
```

```
In [24]: message[0]
```

```
Out [24]: 'w'
```

```
In [24]: # negative indices wraps around the end  
         message[-1] # last character
```

```
Out [24]: '?'
```

String Indexing and Slicing

```
In [23]: message = "what do you like?"
```

```
In [24]: # Access individual characters (zero-based indexing)  
         message[0]
```

```
Out [24]: 'w'
```

```
In [25]: message[0:4] # up to but not including index 4
```

```
Out [25]: 'what'
```

```
In [25]: message[0:7:2] # step size of 2
```

```
Out [25]: 'wa o'
```

String Indexing and Slicing

```
In [26]: message = "python"
```

```
In [26]: # default start index is 0  
        message[:4]
```

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

```
In [27]: # default end index is length of string  
        message[4:]
```

```
Out [27]: 'on'
```

```
In [28]: message[:] # default 0 to end of string
```

```
Out [28]: 'python'
```

String Indexing and Slicing

```
In [26]: message = "python"
```

```
In [24]: # negative indices wraps around the end  
         message[-1] # last character
```

```
Out [24]: 'n'
```

```
In [25]: # all except the last character  
         message[:-1]
```

```
Out [25]: 'pytho'
```

```
In [25]: # negative step size traverses backwards  
         message[::-1]
```

```
Out [25]: 'nohtyp'
```


f-Strings

f-Strings is the new way to format strings in Python. (v 3.6)

```
In [26]: name = "Mike"  
        gpa = 3.2  
        f_str = f"I am {name} with a {gpa} gpa."  
        print(f_str)
```

```
Out [26]: 'I am Mike with a 3.2 gpa.'
```

f-Strings Precision

```
In [26]: import math  
         x = math.pi  
         print(f"{x}")  
         print(f"{x:.2f}")  
         print(f"{x:.3f}")
```

3.141592653589793

3.14

3.142

NoneType

Python includes a special type, the `NoneType`, which has only a single possible value: `None`.

```
In [24]: type(None)
```

```
Out [24]: NoneType
```

You'll see `None` used in many places, but perhaps most commonly it is used as the default return value of a function.

```
In [25]: return_value = print('abc')
```

```
abc
```

```
In [26]: print(return_value)
```

```
None
```

NoneType

You'll see None used in many places, but perhaps most commonly it is used as the default return value of a function.

```
In [25]: return_value = print('abc')
```

abc

```
In [26]: print(return_value)
```

None

References

I) Vanderplas, Jake, A Whirlwind Tour of Python, O'reilly Media.

This book is completely free and can be downloaded online at O'reilly's site.