

Strings

Strings

Three ways to define string literals:

- ▶ with single quotes: 'Ni!'
- ▶ double quotes: "Ni!"
- ▶ Or with triples of either single or double quotes, which creates a multi-line string:

```
1 >>> """I do HTML for them all,  
2 ... even made a home page for my dog."""  
3 'I do HTML for them all,\neven made a home page for my dog.'
```

Strings

Note that the REPL echoes the value with a `\n` to represent the newline character. Use the `print` function to get your intended output:

```
1 >>> nerdy = """I do HTML for them all,  
2 ... even made a home page for my dog."""  
3 >>> nerdy  
4 'I do HTML for them all,\neven made a home page for my dog.'  
5 >>> print(nerdy)  
6 I do HTML for them all,  
7 even made a home page for my dog.
```

That's pretty `nerdy`.

Strings

Choice of quote character is usually a matter of taste, but the choice can sometimes buy convenience. If your string contains a quote character you can either escape it:

```
1 >>> journey = 'Don\'t stop believing.'
```

or use the other quote character:

```
1 >>> journey = "Don't stop believing."
```

► How does Python represent the value of the variable `journey` ?

String Operations

Because strings are sequences we can get a string's length with

`len()`:

```
1 >>> i = "team"
2 >>> len(i)
3 4
```

and access characters in the string by index (offset from beginning – first index is 0) using `[]`:

```
1 >>> i[1]
2 'e'
```

Note that the result of an index access is a string:

```
1 >>> type(i[1])
2 <class 'str'>
3 >>> i[3] + i[1]
4 'me'
5 >>> i[-1] + i[1] # Note that a negative index goes from the end
6 'me'
```

- ▶ What is the index of the first character of a string?
- ▶ What is the index of the last character of a string?

String Slicing

`[:end]` gets the first characters up to but not including `end`

```
1 >>> al_gore = "manbearpig"
2 >>> al_gore[:3]
3 'man'
```

`[begin:end]` gets the characters from `begin` up to but not including `end`

```
1 >>> al_gore[3:7]
2 'bear'
```

`[begin:]` gets the characters from `begin` to the end of the string

```
1 >>> al_gore[7:]
2 'pig'
3 >>>
```

- What is the relationship between the ending index of a slice and the beginning index of a slice beginning right after the first slice?

String Methods

`str` is a class (you'll learn about classes later) with many methods (a method is a function that is part of an object). Invoke a method on a string using the dot operator.

`str.find(substr)` returns the index of the first occurrence of `substr` in `str`

```
1 >>> 'foobar'.find('o')
2 1
```

- ▶ Write a string slice expression that returns the username from an email address, e.g., for 'bob@aol.com' it returns 'bob'.
- ▶ Write a string slice expression that returns the host name from an email address, e.g., for 'bob@aol.com' it returns 'aol.com'.

Conclusion

Your turn:

- ▶ Exercise 1