

Strings



Strings



Basic Data Types

Strings

Integers

Booleans

Floats



Strings

"STRINGS OF CHARACTERS"

Strings are a textual datatype and must be wrapped in quotes

chicken nuggets for ever!

chicken nuggets for ever!



```
color = "Magenta"
```

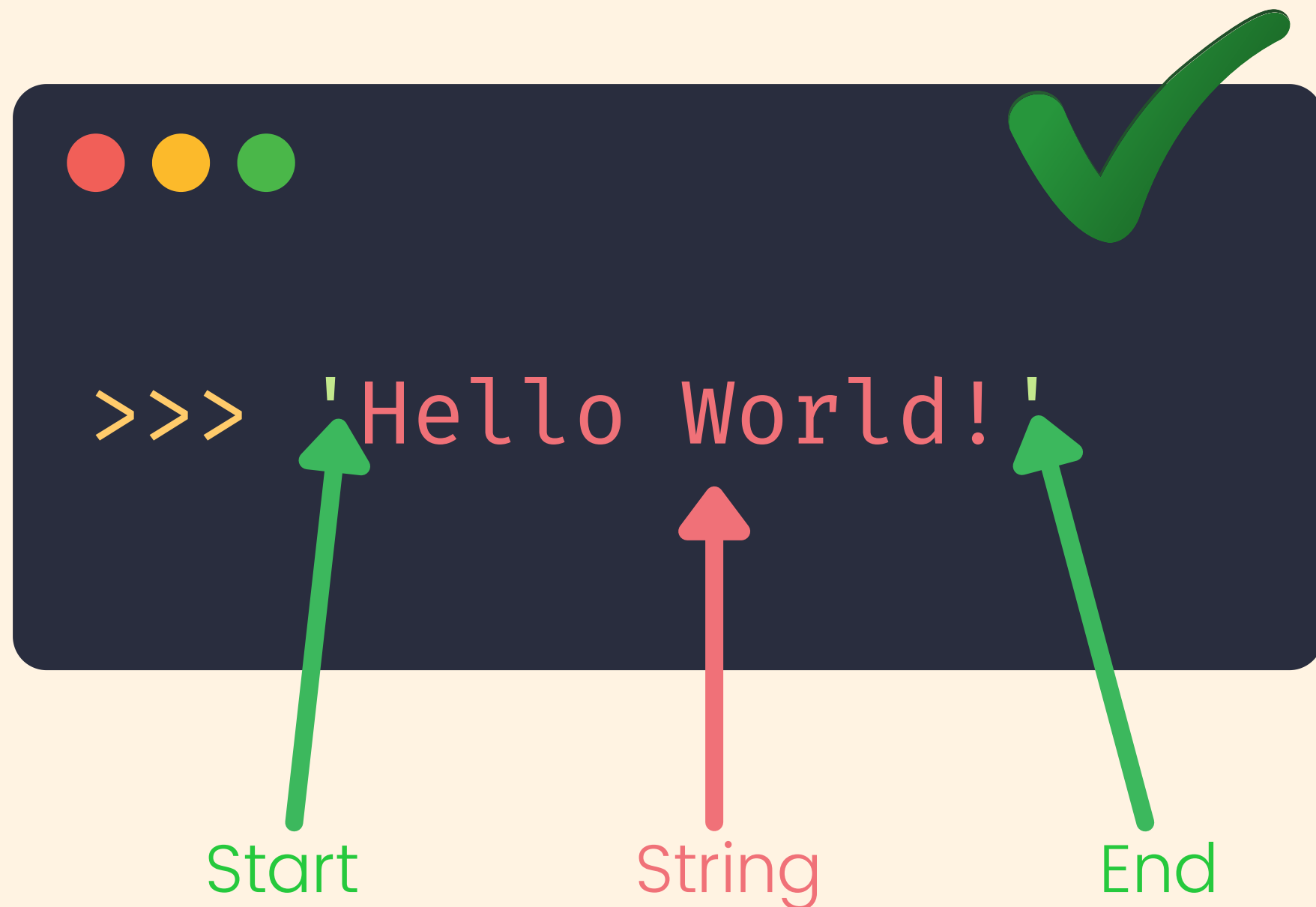


```
twitter_handle = '@POTUS'
```



```
url = "www.reddit.com/r/formula1/"
```

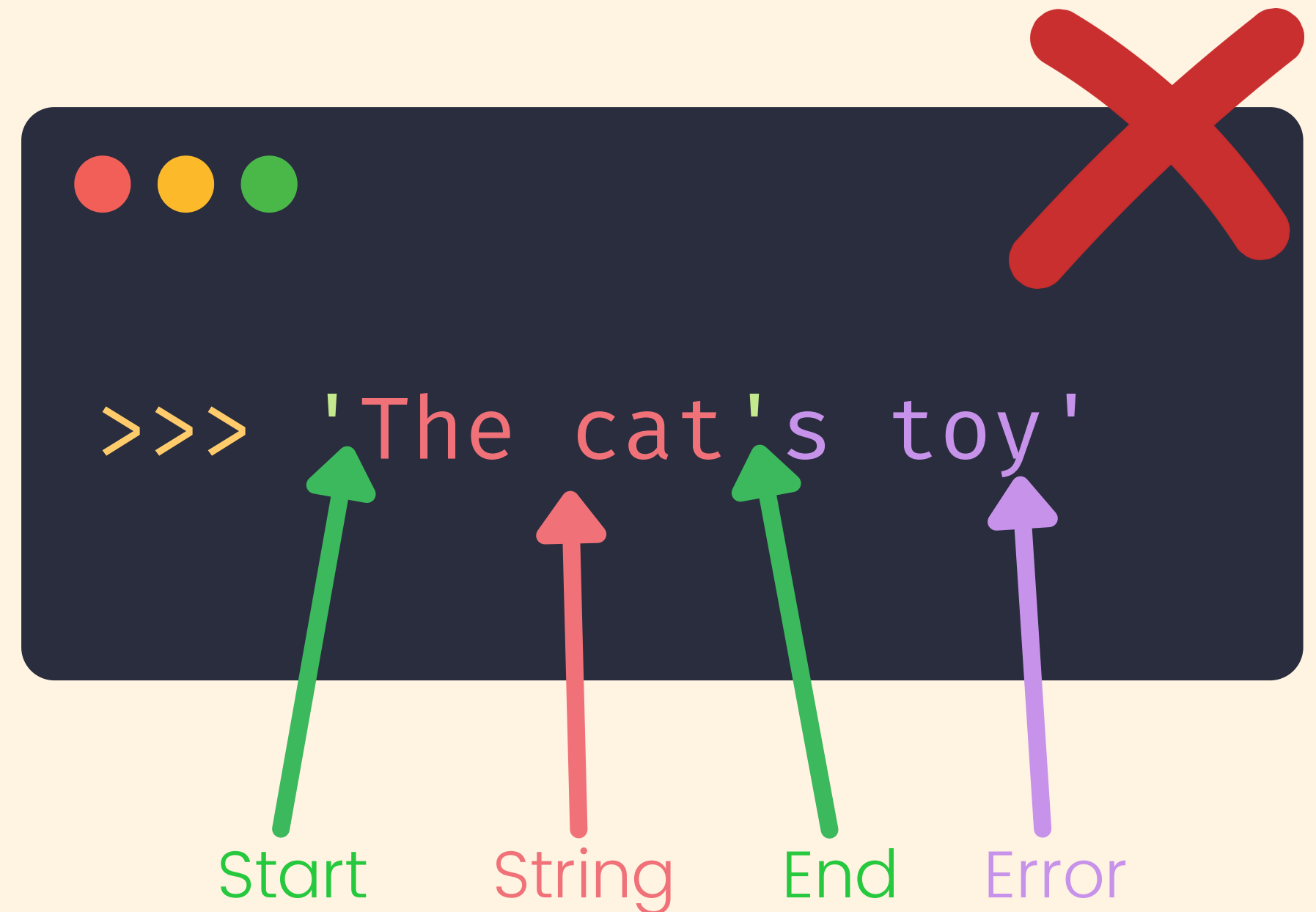
Quotes



```
>>> 'Hello World!'
```

Start String End

A terminal window with a dark blue background and three colored window control buttons (red, yellow, green) in the top-left corner. A large green checkmark is in the top-right corner. The prompt '>>>' is in yellow. The string 'Hello World!' is in pink. A green arrow points from the label 'Start' to the opening quote, a red arrow points from 'String' to the text, and another green arrow points from 'End' to the closing quote.

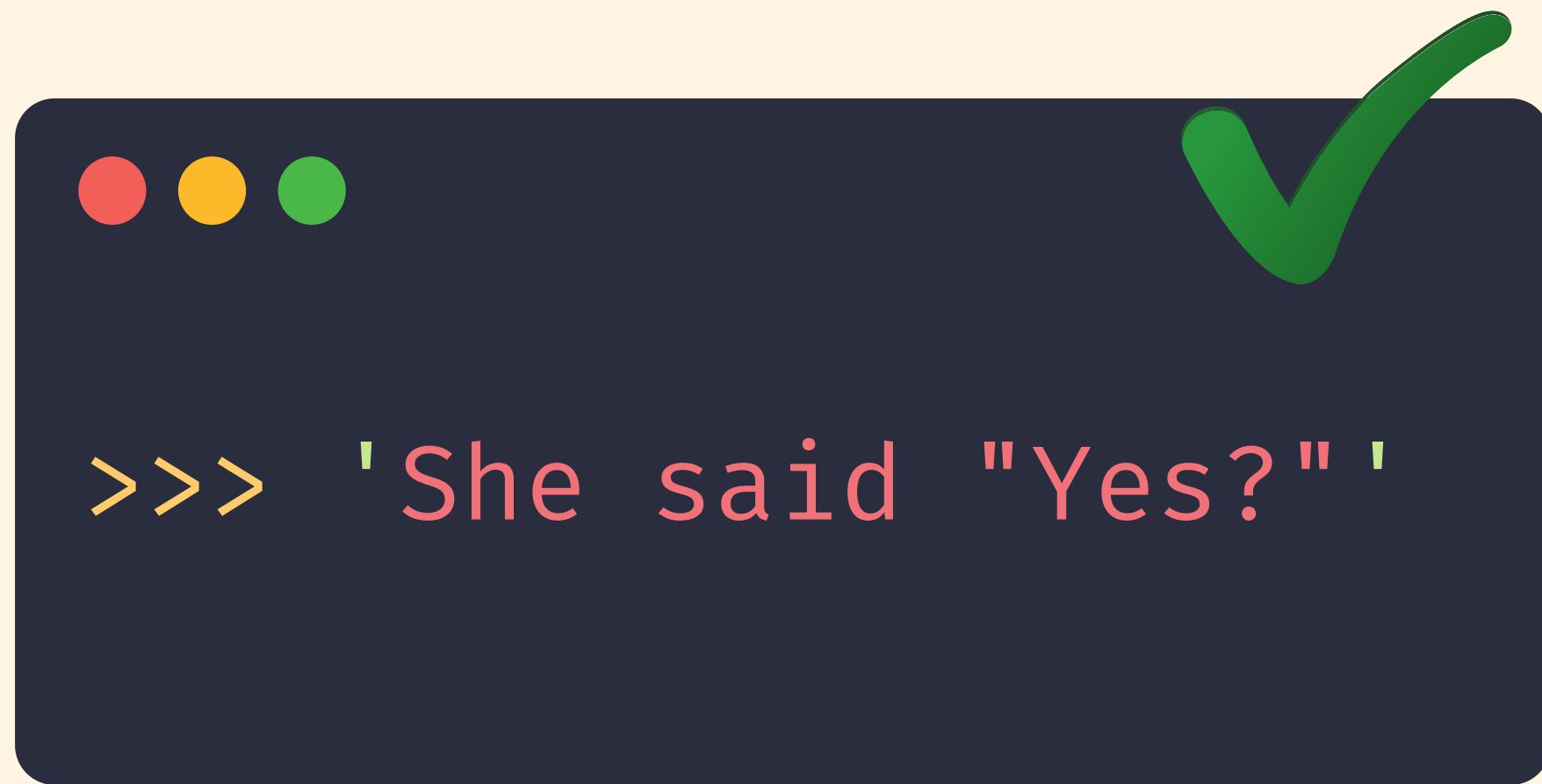


```
>>> 'The cat's toy'
```

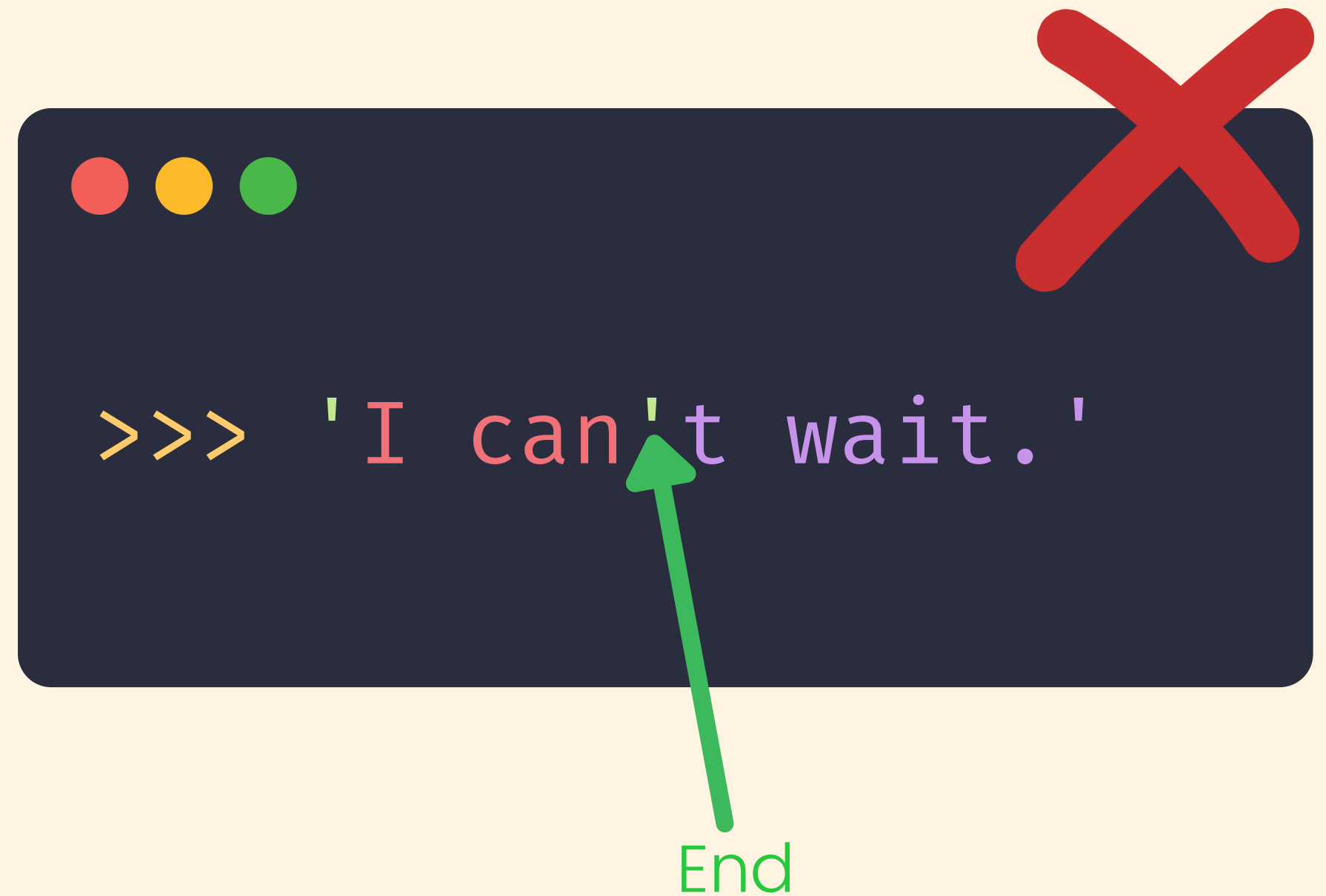
Start String End Error

A terminal window with a dark blue background and three colored window control buttons (red, yellow, green) in the top-left corner. A large red X is in the top-right corner. The prompt '>>>' is in yellow. The string 'The cat's toy' is in pink. A green arrow points from the label 'Start' to the opening quote, a red arrow points from 'String' to the text, a green arrow points from 'End' to the closing quote, and a purple arrow points from 'Error' to the apostrophe.

Single Quotes

A terminal window with a dark blue background and three colored window control buttons (red, yellow, green) in the top-left corner. A large green checkmark is positioned in the top-right corner of the window. The terminal prompt is '>>>' in yellow, followed by the string 'She said "Yes?"' in pink, enclosed in single quotes.

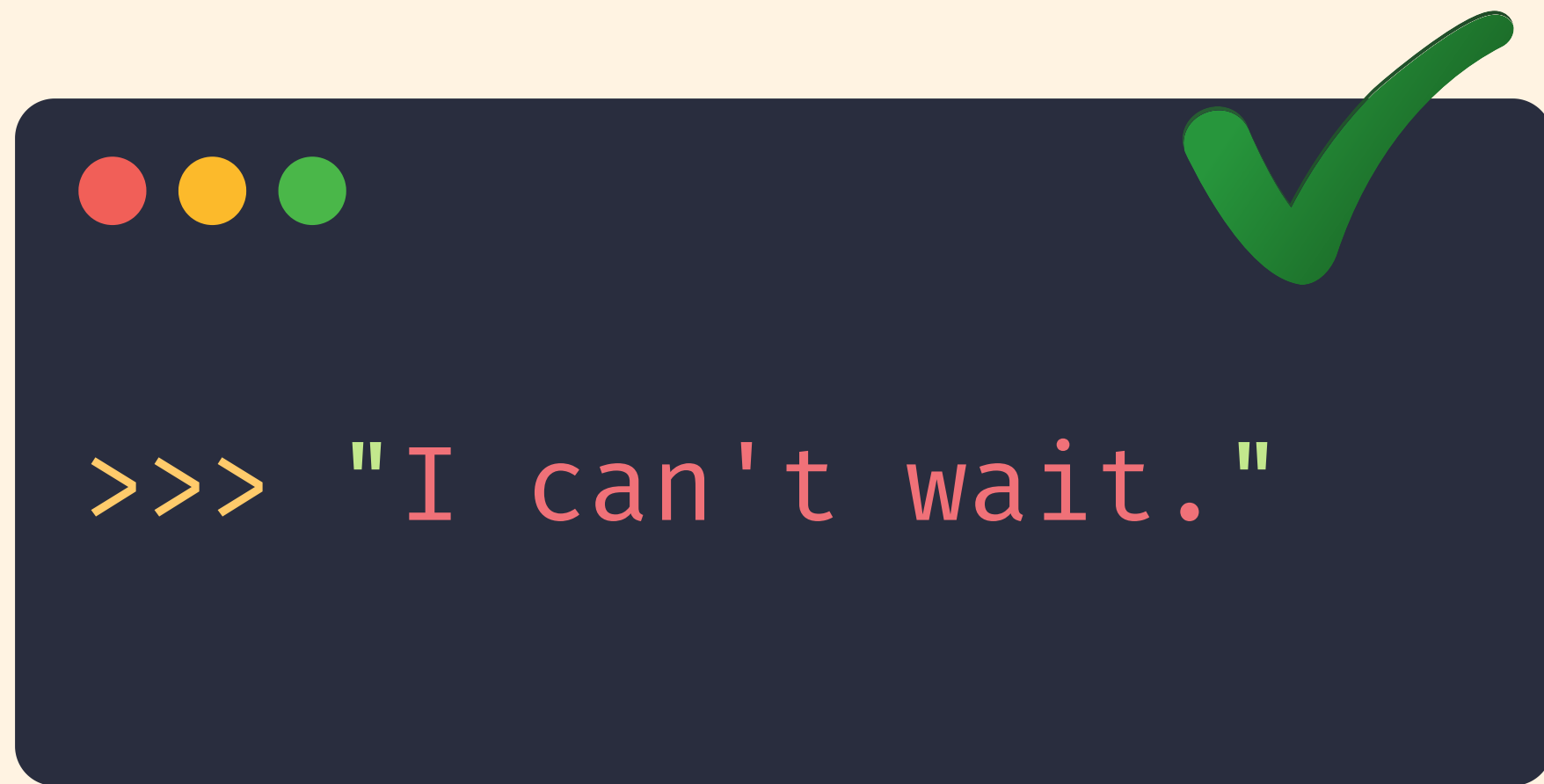
```
>>> 'She said "Yes?"'
```

A terminal window with a dark blue background and three colored window control buttons (red, yellow, green) in the top-left corner. A large red X is positioned in the top-right corner of the window. The terminal prompt is '>>>' in yellow, followed by the string 'I can't wait.' in purple, enclosed in single quotes. A green arrow points from the word 'End' below the terminal to the single quote character after the word 'can' in the string.

```
>>> 'I can't wait.'
```

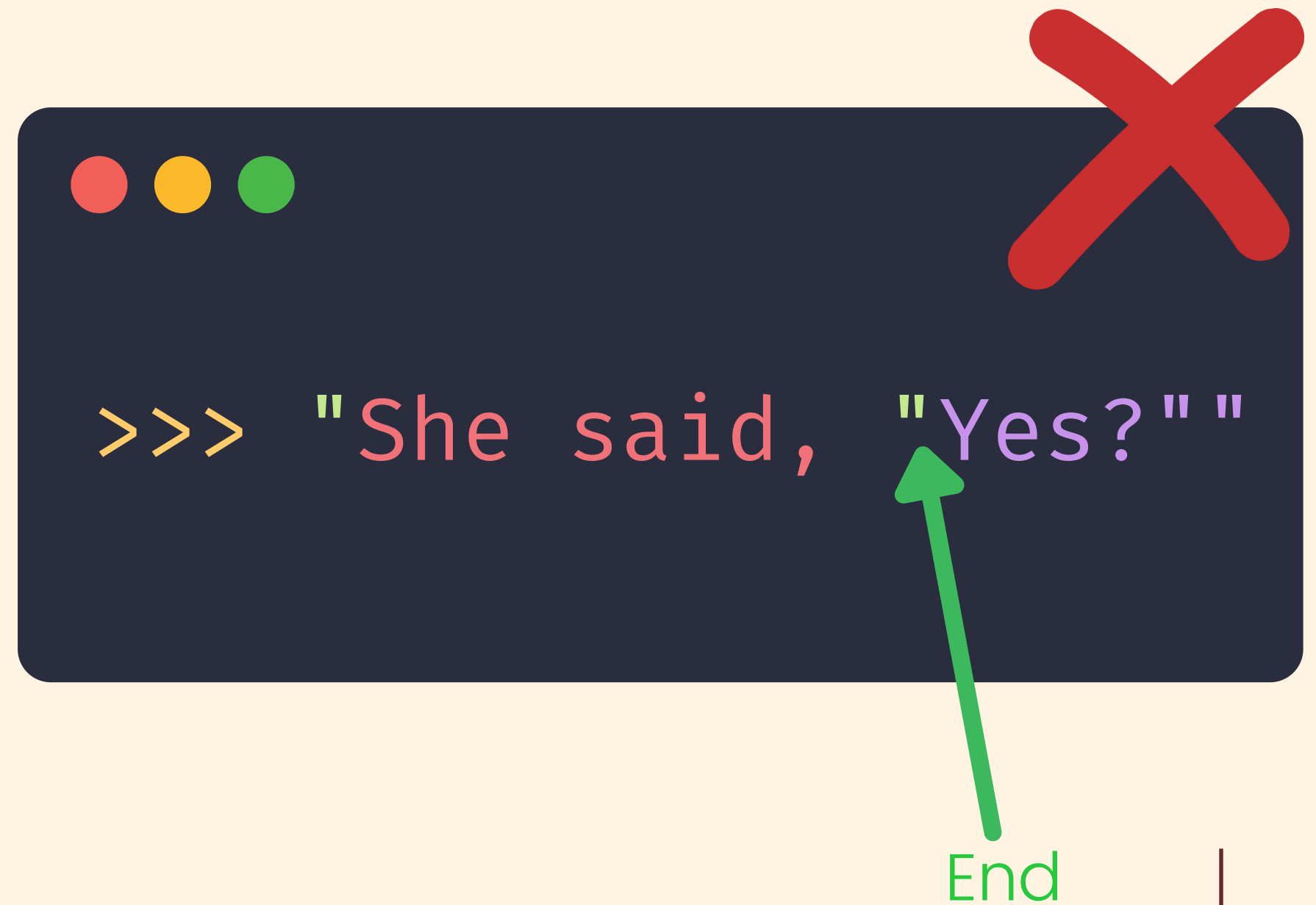
End

Double Quotes



```
>>> "I can't wait."
```

A terminal window with a dark blue background and three colored window control buttons (red, yellow, green) in the top-left corner. A large green checkmark is positioned in the top-right corner of the window. The text inside the terminal is `>>> "I can't wait."` in a light pink font.



```
>>> "She said, "Yes?""
```

A terminal window with a dark blue background and three colored window control buttons (red, yellow, green) in the top-left corner. A large red X is positioned in the top-right corner of the window. The text inside the terminal is `>>> "She said, "Yes?""` in a light pink font. A green arrow points from the word "End" below the terminal to the closing double quote of the inner string. A small black arrow points downwards from the bottom right of the terminal area.

Triple Quotes

Single Or Double



```
>>> '''"Colt's sourdough bread is  
the best," Paul Hollywood stated.'''
```

Print

The `print()` function prints out any values we pass to it to "standard output". It does not return anything.



```
>>> print("hello")
```

Escape Characters



Newline - `\n`

Double Quote - `\"`

Single Quote - `\'`

Tab - `\t`

Backslash - `\\`

Concatenation

We can concatenate strings together by using the plus sign. No space will be added between them.

```
>>> 'pan' + 'cake'  
'pancake'
```

Multiplication

We can also multiply a string by a number, which will repeat that string.

```
>>> 'ha' * 4  
'hahaha'
```

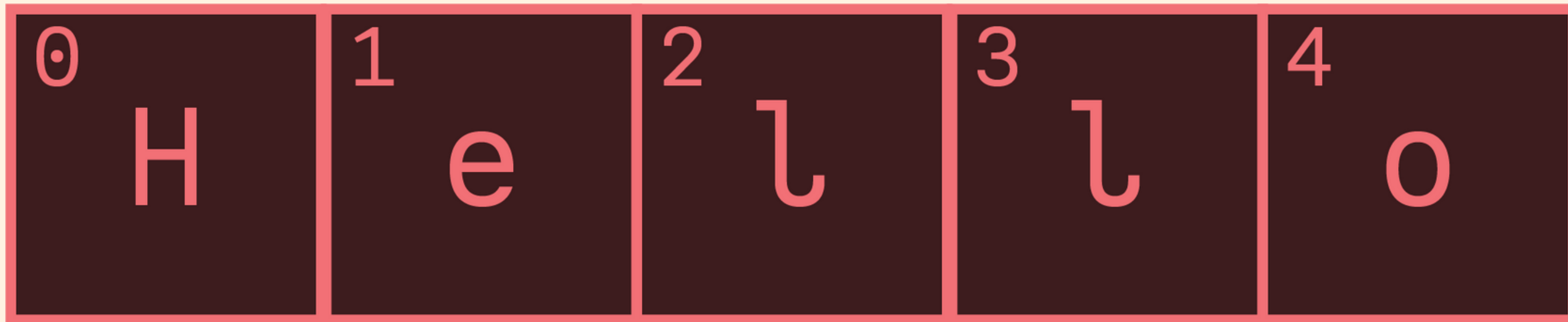
None

None is a special value in Python that denotes the **lack of value**. It is not the same as zero or an empty string (those are still values).

```
>>> user = None
```

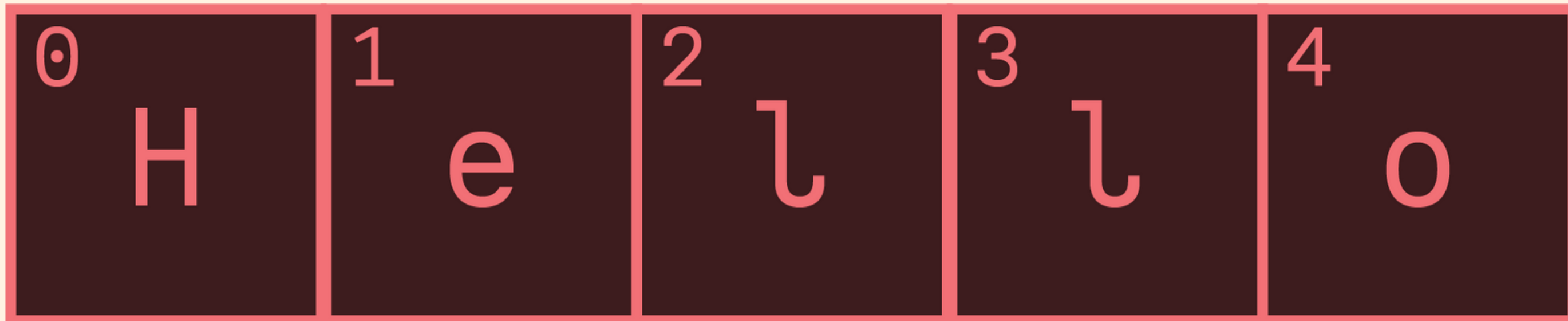


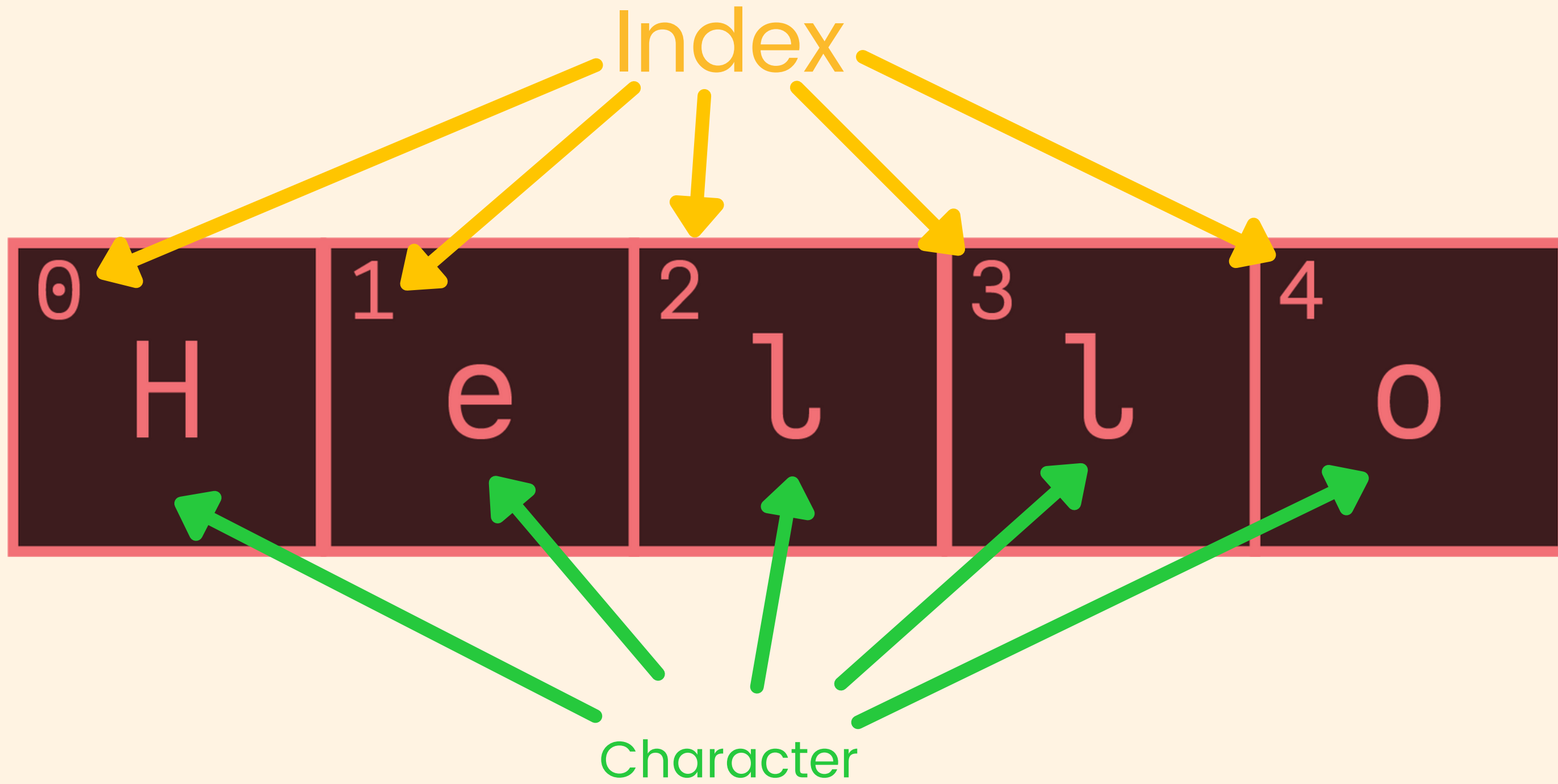
Strings Are Ordered





Strings Are Indexed

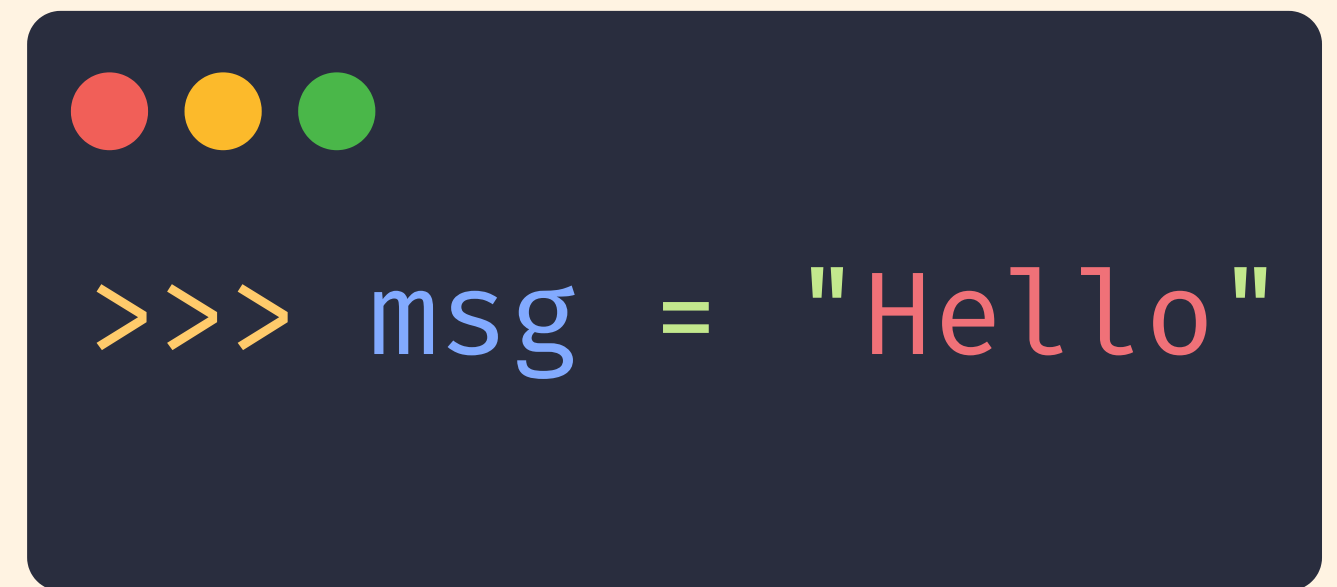






What is a name?

What is a name?
That which we
call a string
by any other name
would still say the same thing.



```
>>> msg = "Hello"
```



How Variables Work

Your Code

```
>>> msg = "Hello"
```

Names

msg



Objects

Hello

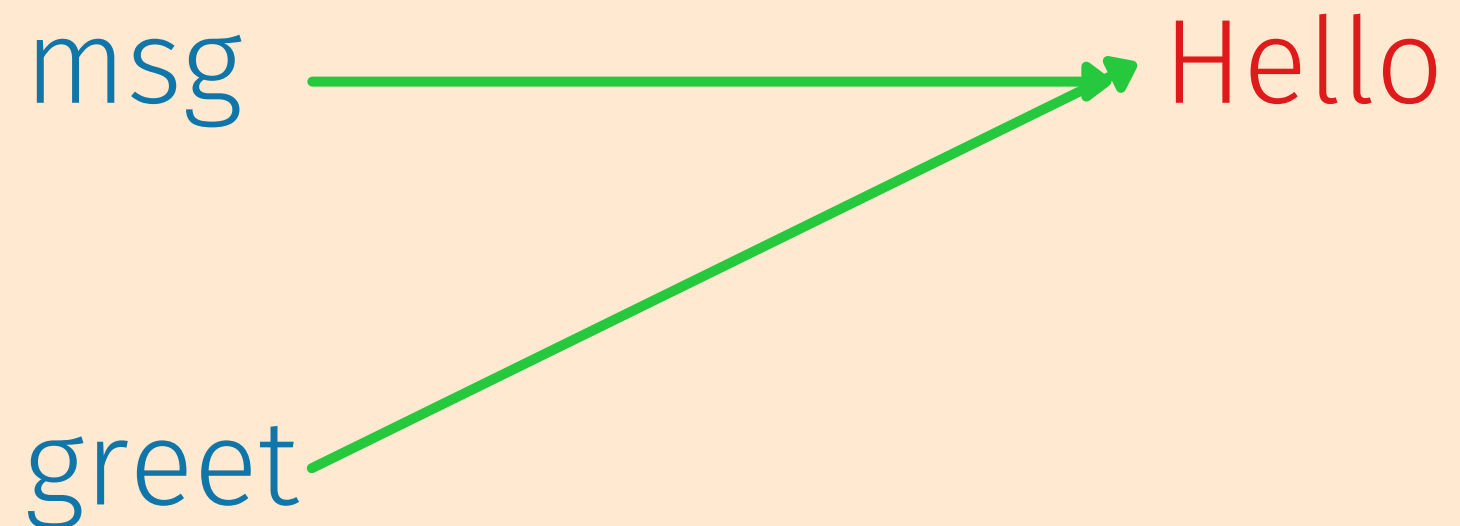
How Variables Work

Your Code

```
• • •  
>>> msg = "Hello"  
>>> greet = "Hello"
```

Names

Objects



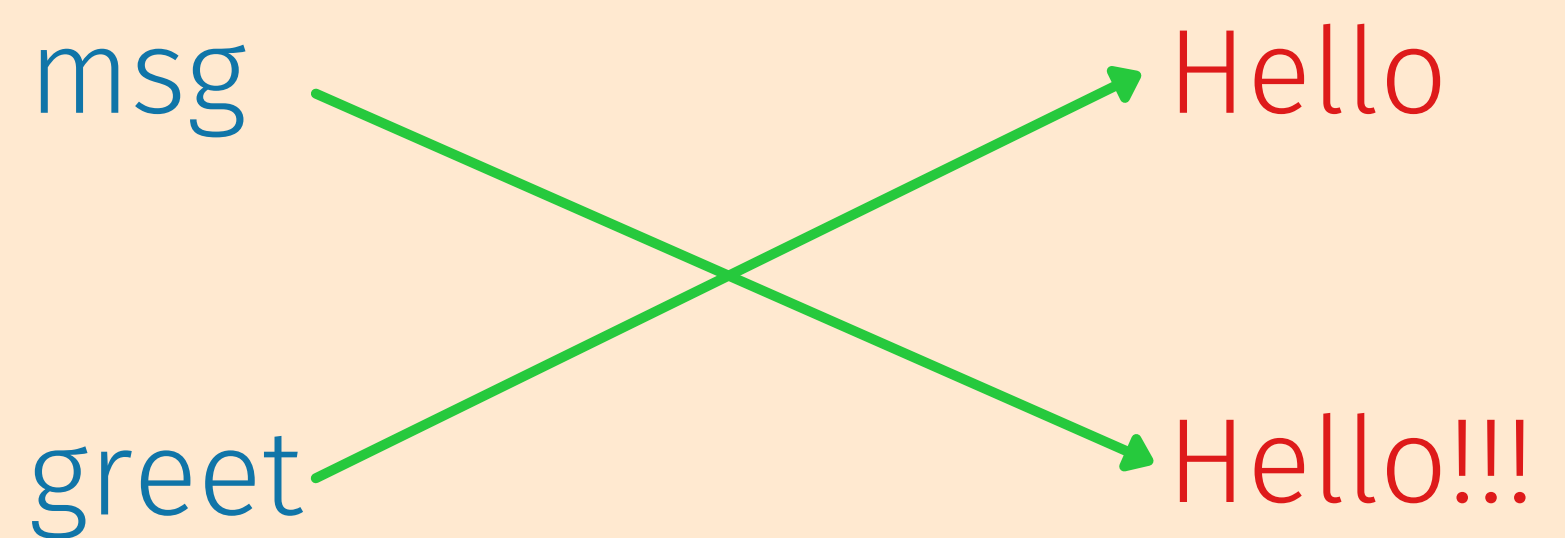
How Variables Work

Your Code

```
>>> msg = "Hello"  
>>> greet = "Hello"  
>>> msg = "Hello!!!"
```

Names

Objects



Indexes

0	1	2	3	4	5	6	7	8
I		<	3		C	a	t	s

```
>>> msg = "I <3 Cats"
>>> msg[0]
'I'
>>> msg[5]
'C'
```



0 1 2 3 4 5 6 7 8

0	1	2	3	4	5	6	7	8
I		<	3		C	a	t	s

0 1 2 3 4 5 6 7 8

⁰ I	¹	² <	³ 3	⁴	⁵ C	⁶ a	⁷ t	⁸ s
-------------------	--------------	-------------------	-------------------	--------------	-------------------	-------------------	-------------------	-------------------

-9 -8 -7 -6 -5 -4 -3 -2 -1

Slices

0	1	2	3	4	5	6	7	8
I		<	3		C	a	t	s

Start

Stop

```
>>> msg[2:6]
'<3 C'
```



Slices with a Step

0	1	2	3	4	5	6	7	8
I		<	3		C	a	t	s

