BOOLEANS + COMPARISONS

Basic Data Types

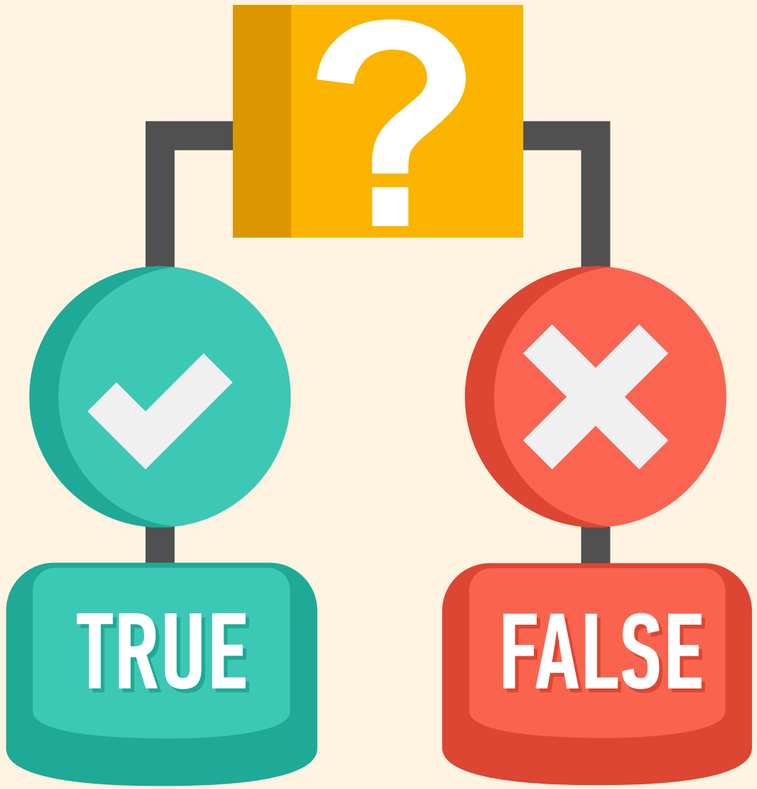
Strings

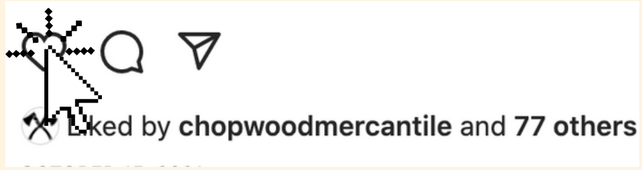
Integers

Booleans

Floats

Decision Making

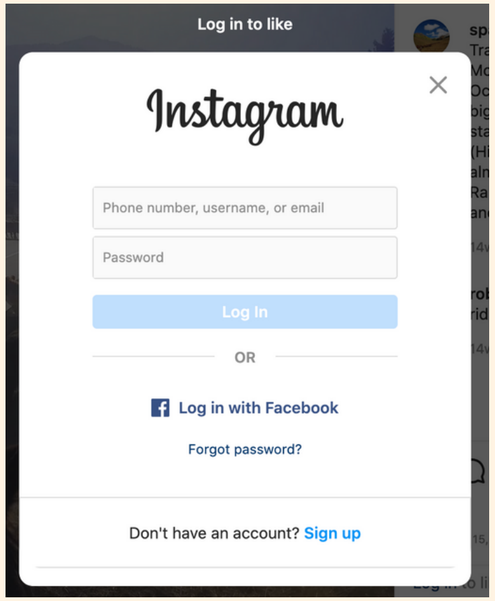




Is the user logged in?

no

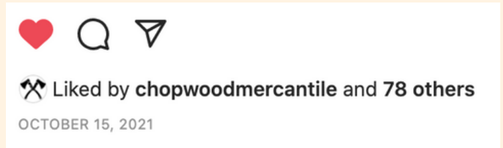
yes



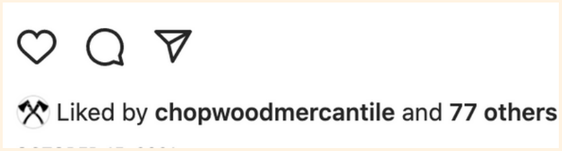
Has the user already liked the photo?

no

yes



Like the photo



un-like the photo

Potential Decisions

Is the game over?

Does the user have any guesses left?

Is 'S' in the target word?

Is 'O' in the target word?

Is 'O' in the correct location?

Is 'U' in the target word?

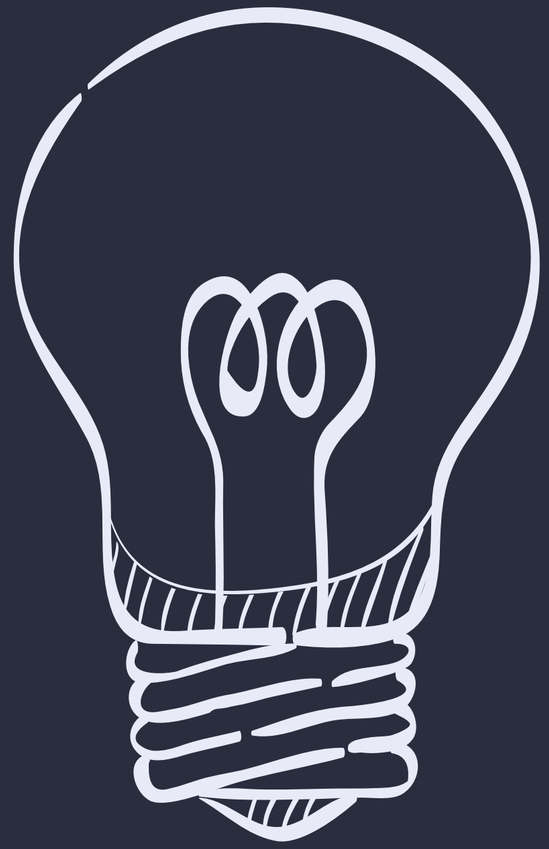
Is 'N' in the target word?

Is 'N' in the correct location?

Is 'D' in the target word?

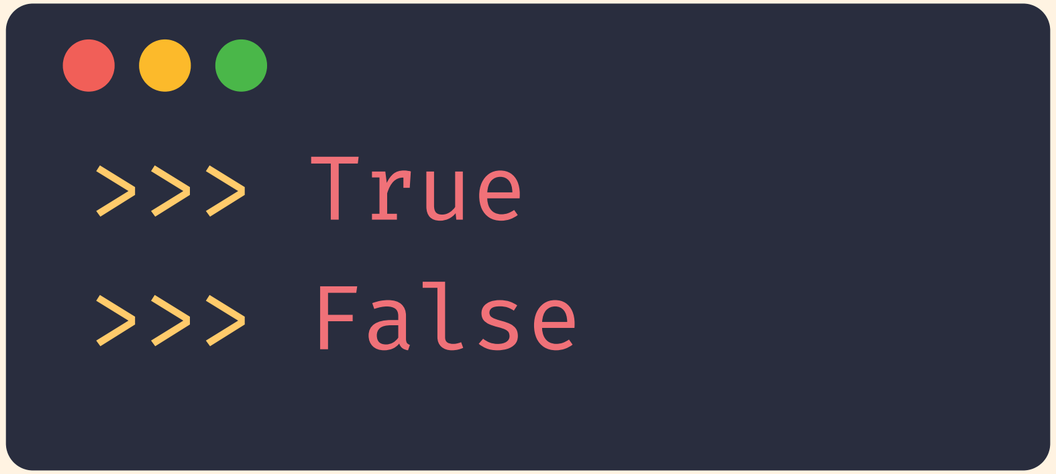
ON

OFF

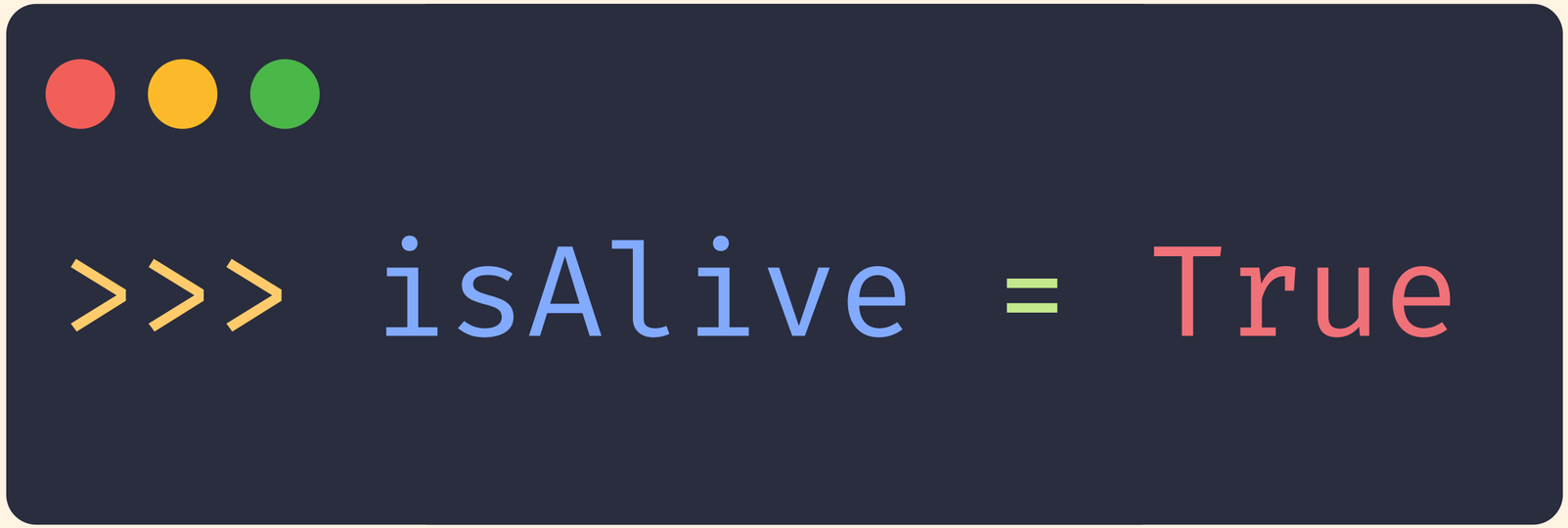


Booleans

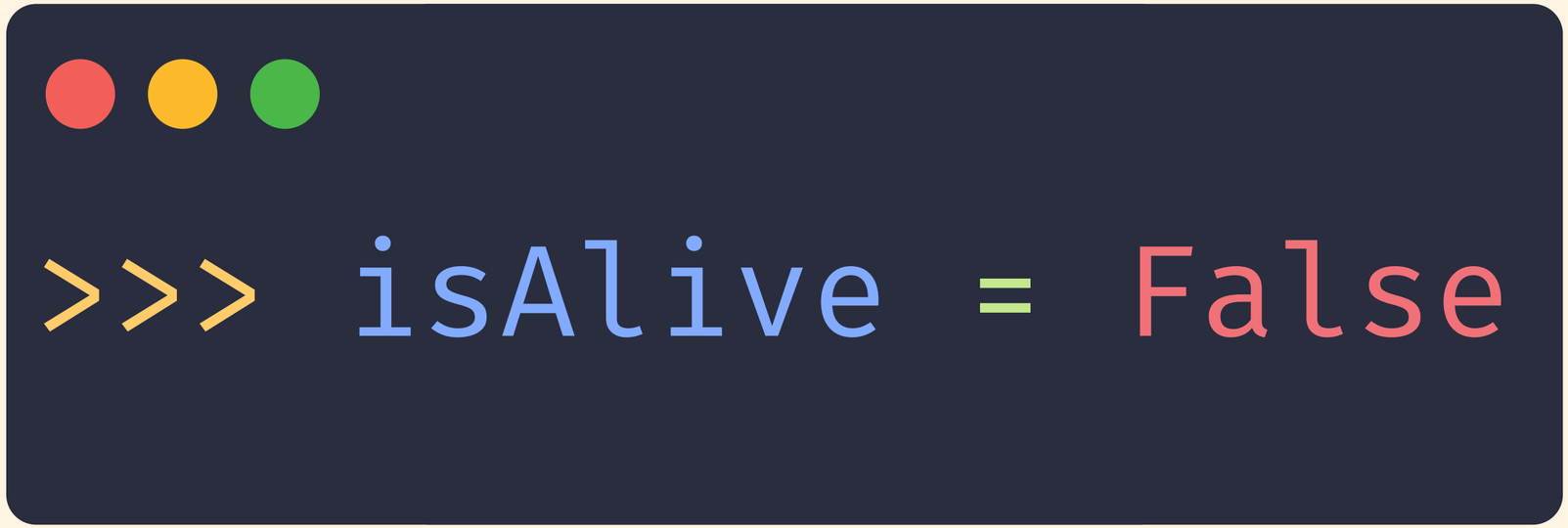
Booleans are another basic Python type. There are only two possible values: **True and False.** Notice the capitalization!!



Booleans



Booleans



Operators

Operators are special characters in Python that perform operations on value(s). Below are some of the most common:

+	*	>	<=	and	is	=	*=
-	/	<	==	or	in	+=	/=
**	%	>=	!=	not	!=	-=	|=

Comparisons

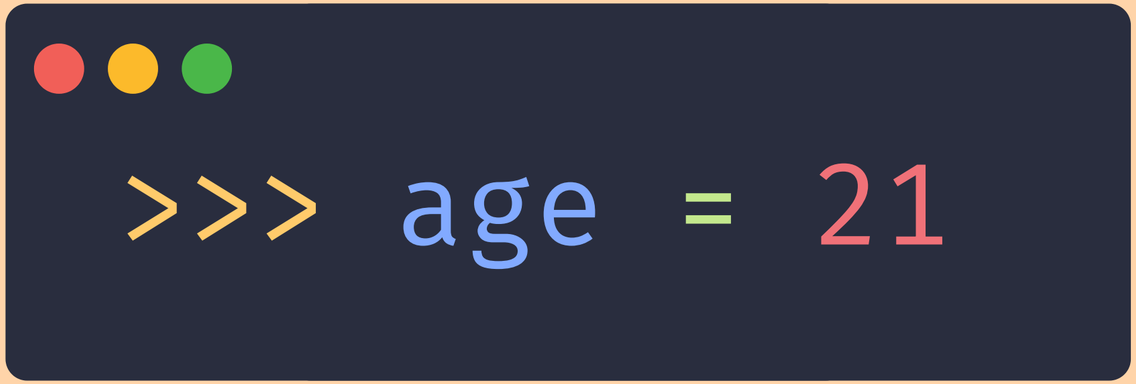
>	Greater Than
<	Less Than
>=	Greater Than Or Equal To
<=	Less Than Or Equal To

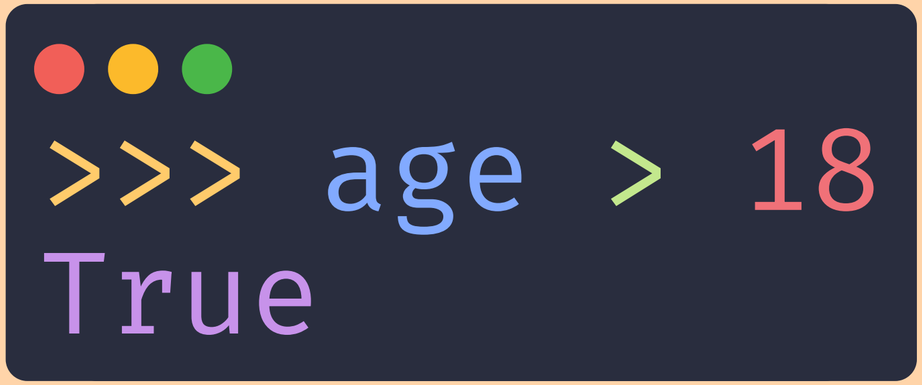
`a > b` Truthy if a is greater than b

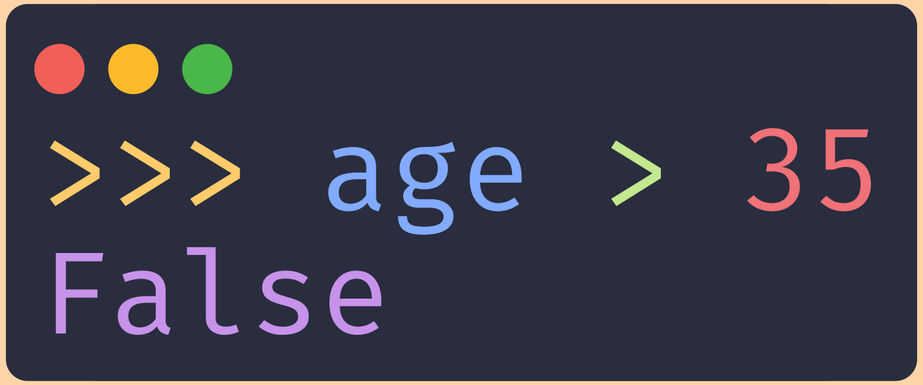
`a < b` Truthy if a is less than b

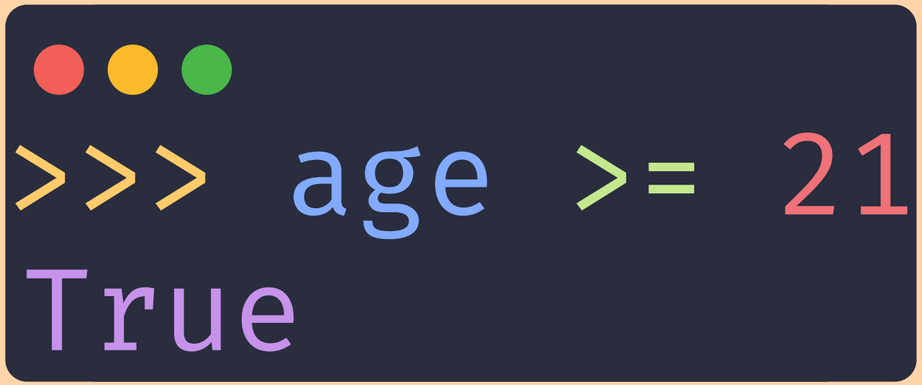
`a >= b` Truthy if a is greater than or equal to b

`a <= b` Truthy if a is less than or equal to b


>>> age = 21

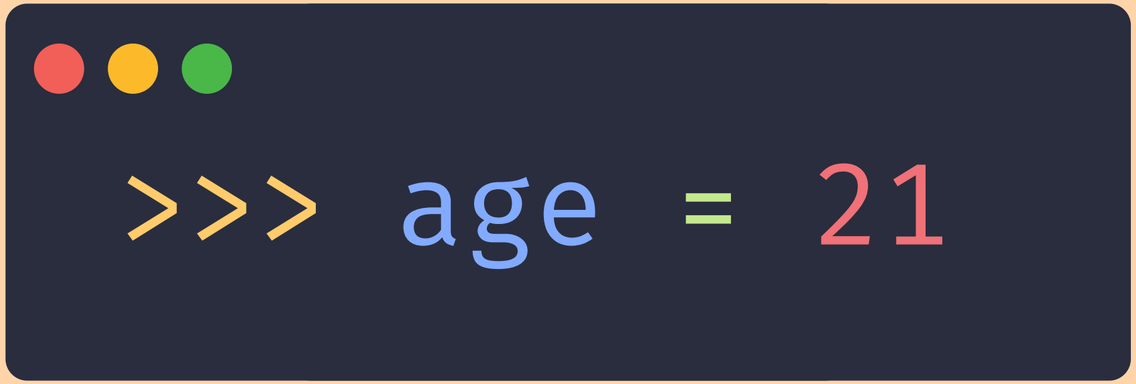

>>> age > 18
True

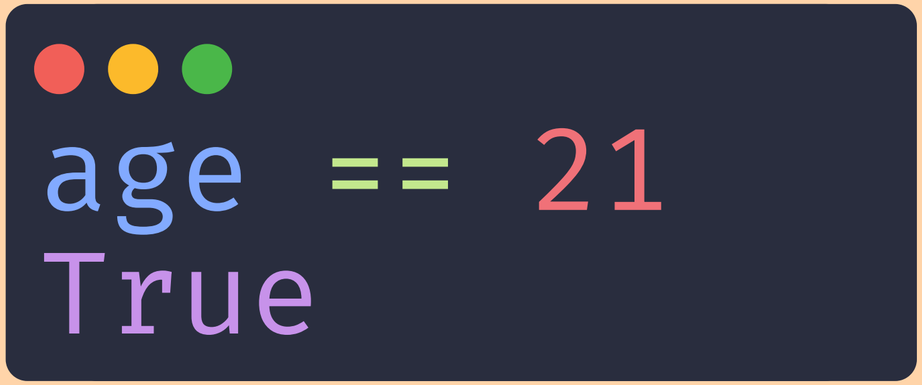

>>> age > 35
False

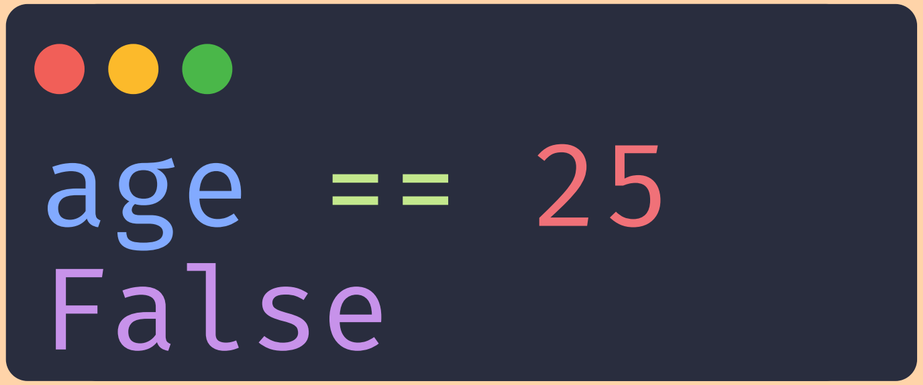

>>> age >= 21
True

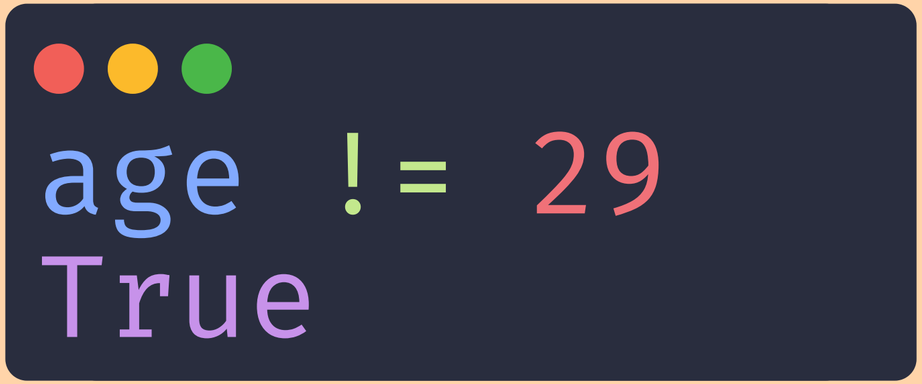
Comparisons

`==`	Equal To
`!=`	Not Equal To


>>> age = 21


age == 21
True

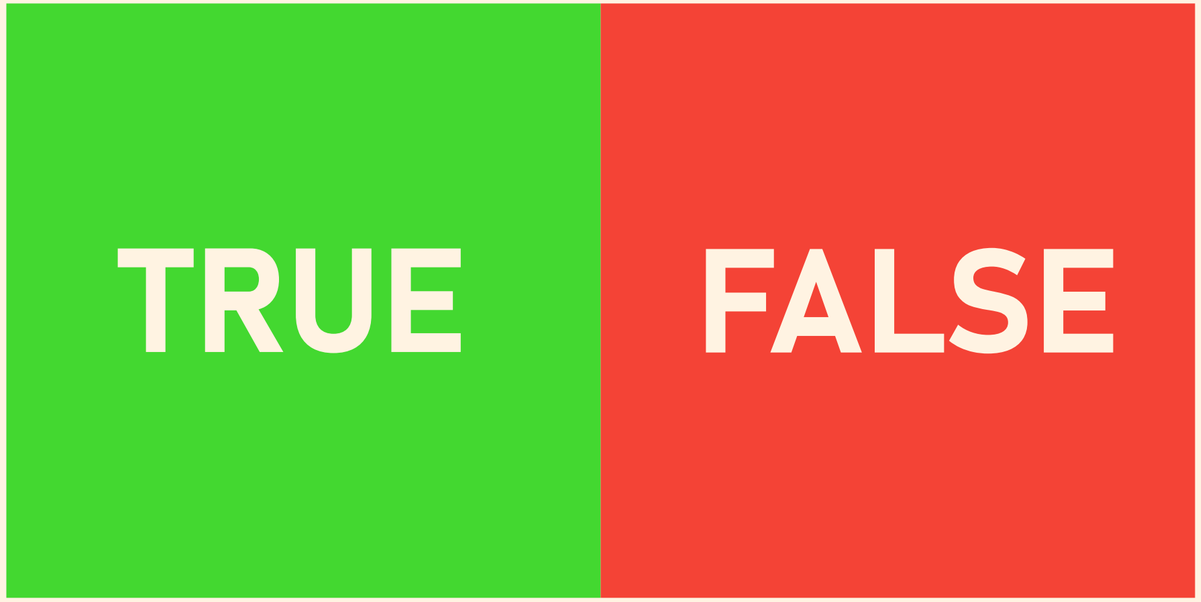

age == 25
False


age != 29
True

Identity

`is`	Evaluates to True if a and b both refer to the same object in memory
`is not`	Evaluates to True if a and b do NOT refer to the same object in memory

Every value is inherently Truth-y or False-y in Python



False-y

False

0.0

0

None

range(0)

Empty Strings:
''
""
''''''
""""""""

Empty Data Structures:
[ ]
( )
{ }
set()

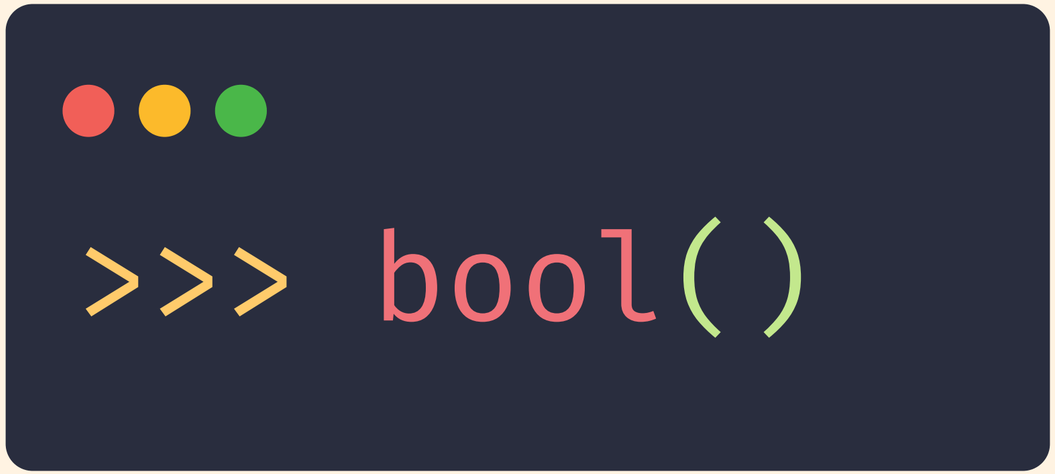
Truthy

Everything Else!

bool( )

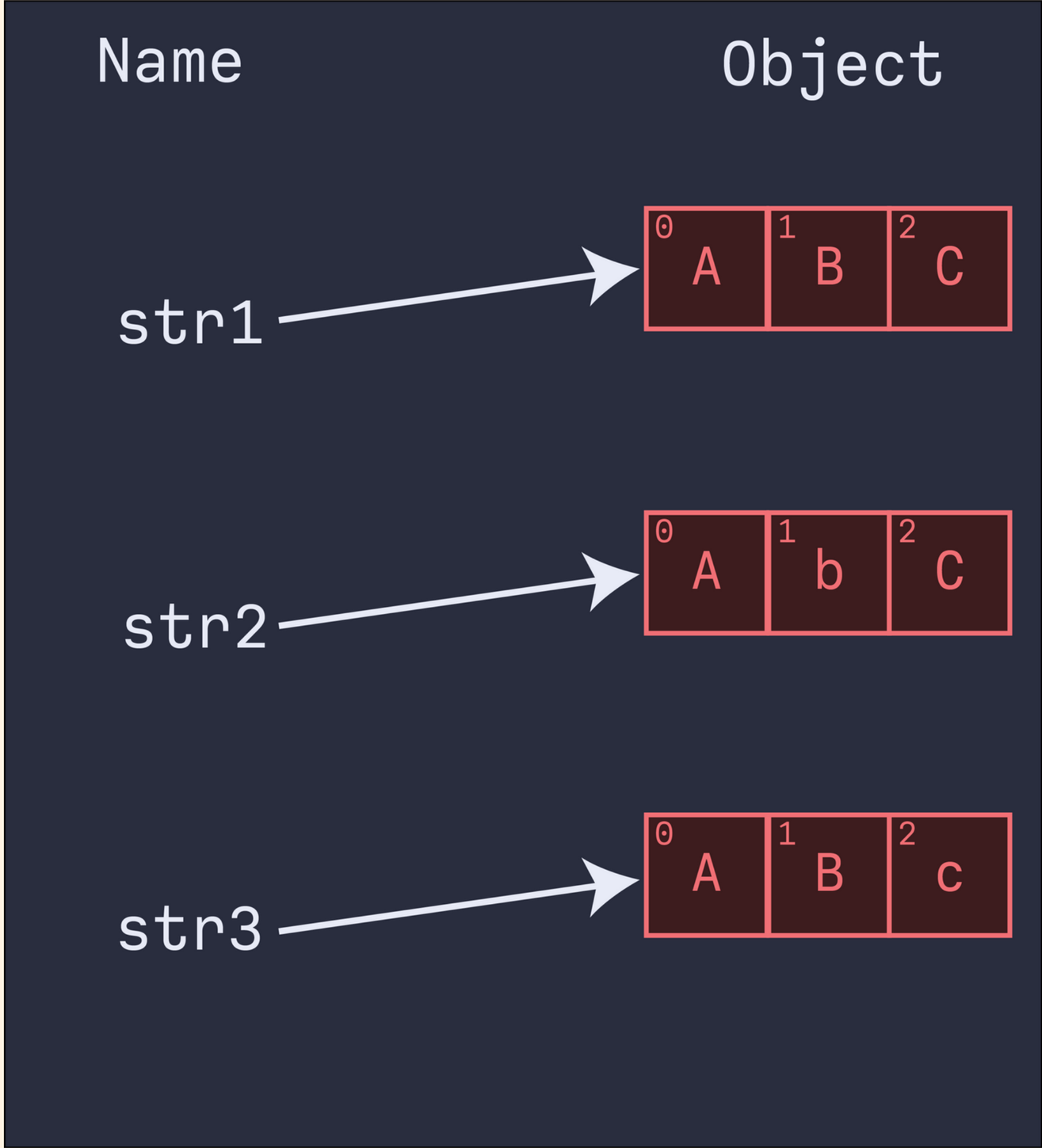
Just as we can use **int**( ), **float**( ), and **str**( ) to cast values, we can use **bool**() to cast a value to a Boolean.

This is one way to determine whether Python considers a value to be Truth-y or False-y



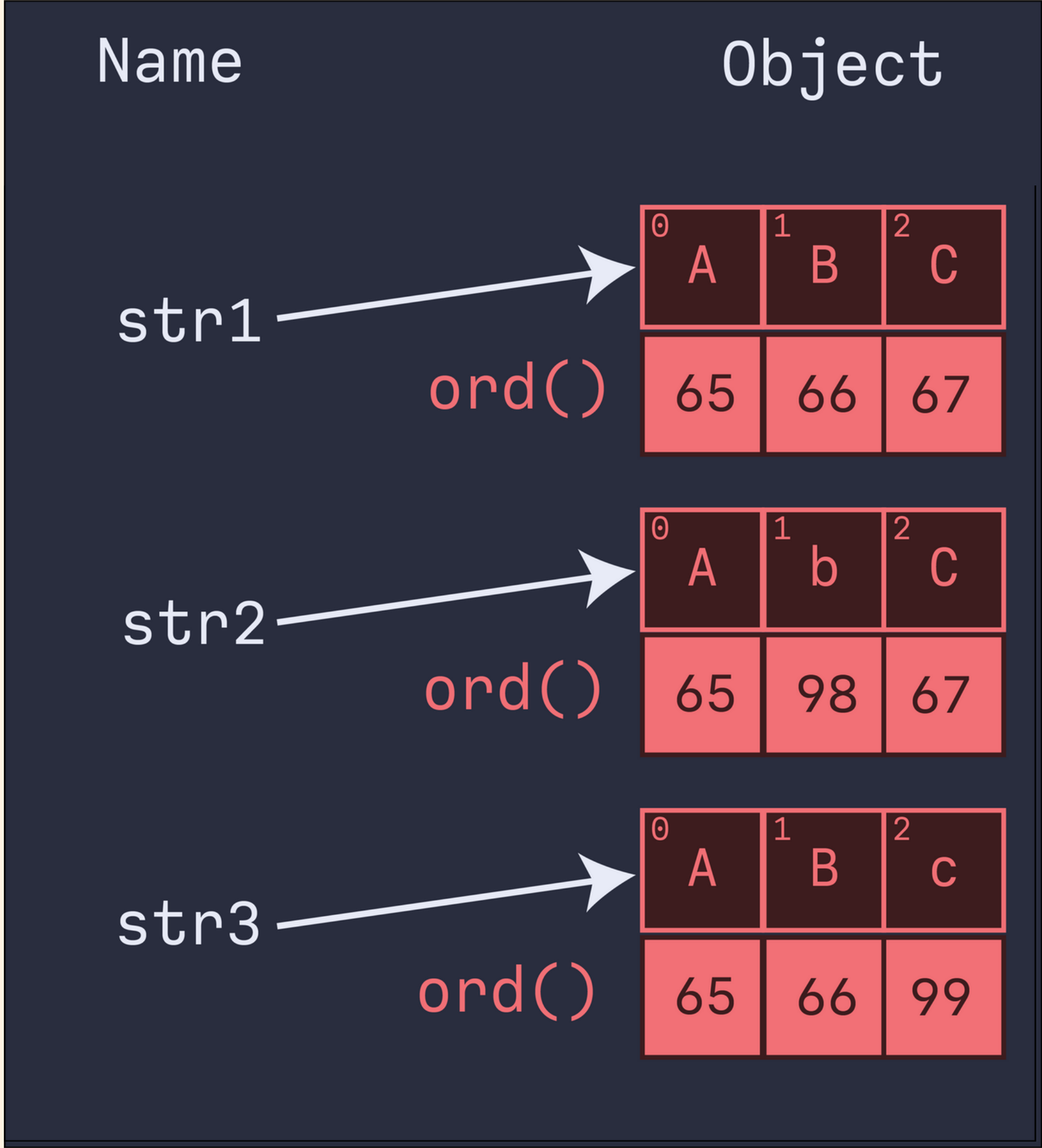
String Comparison

```
>>> str1='ABC'

>>> str2='AbC'

>>> str3='ABc'
```



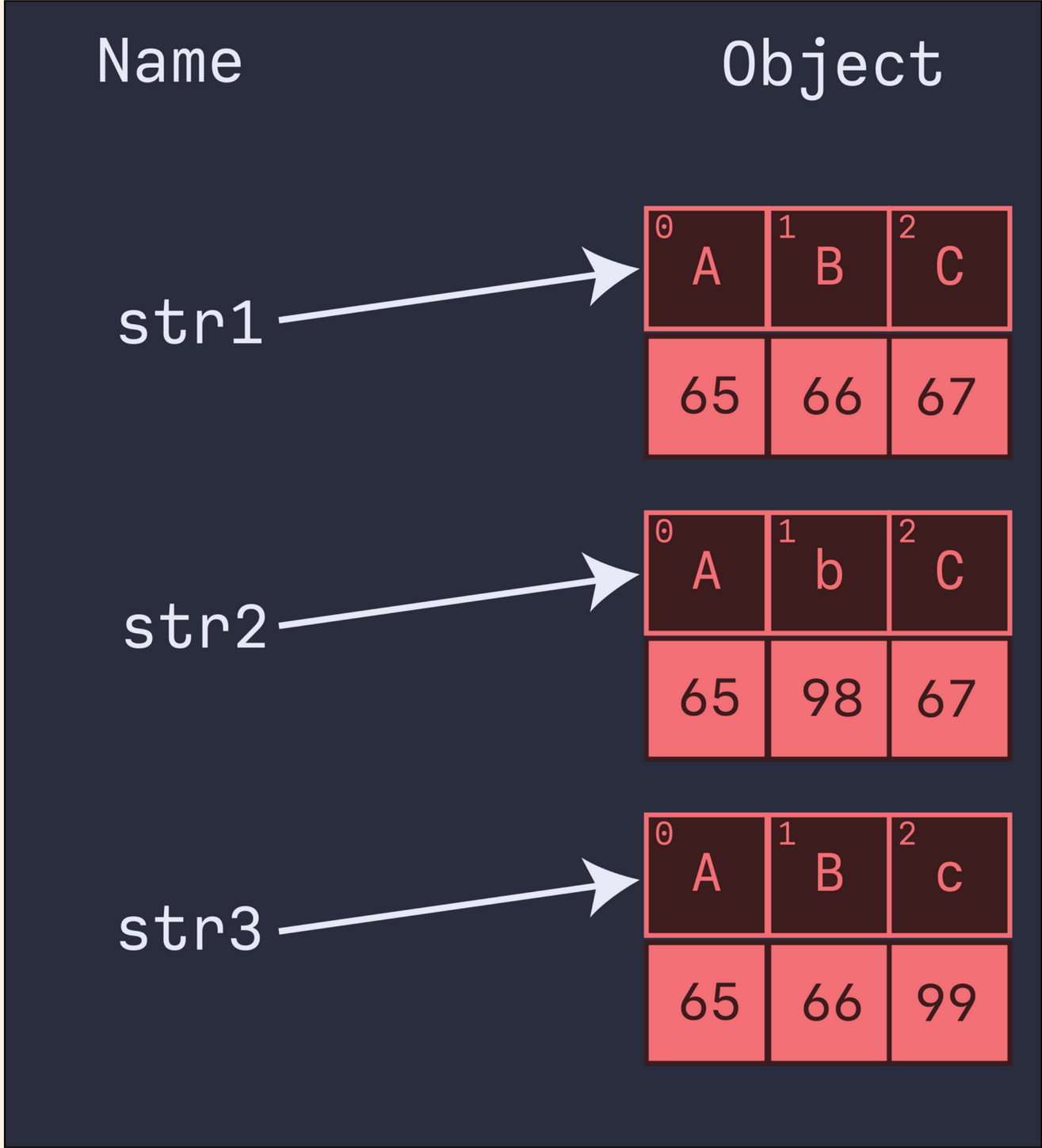
String Comparison

```
>>> ord('A')
65
>>> ord('B')
66
>>> ord('C')
67
>>> ord('b')
98
>>> ord('c')
99
```



String Comparison

```
>>> str1 > str2
False

>>> str2 > str3
True

>>> str1 > str3
False
```



logical and

The **and** operator will evaluate to True only if both the left and right sides evaluate to True.

```
'a' == 'a' and 1 < 5
True
```

logical and

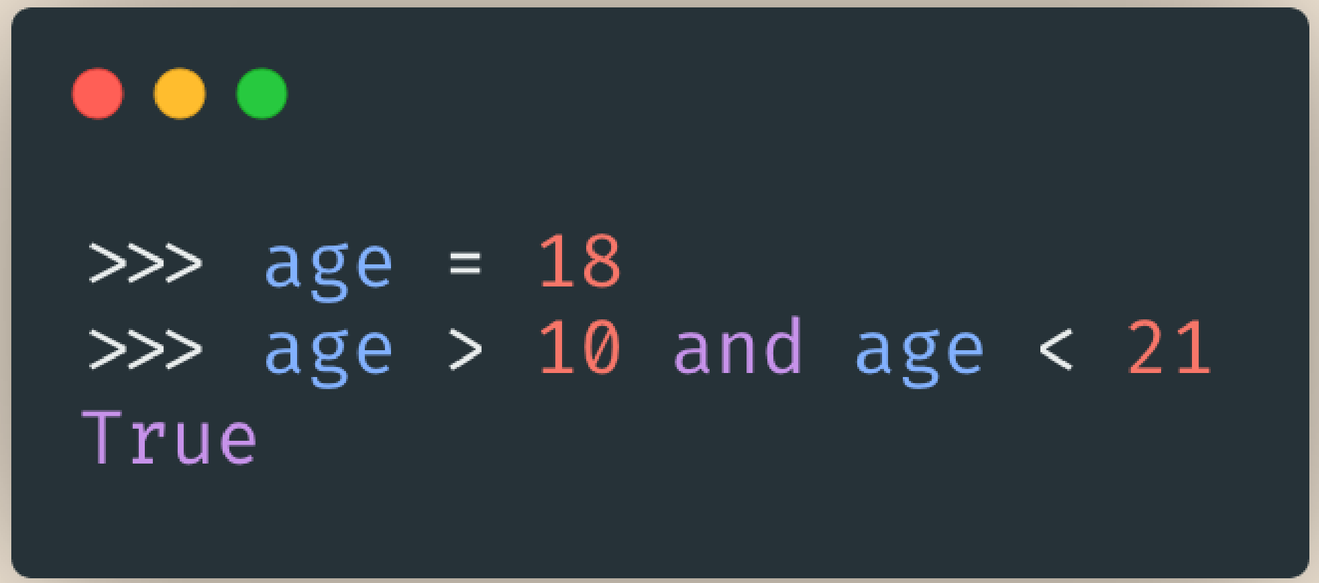
The **and** operator will evaluate to True only if both the left and right sides evaluate to True.

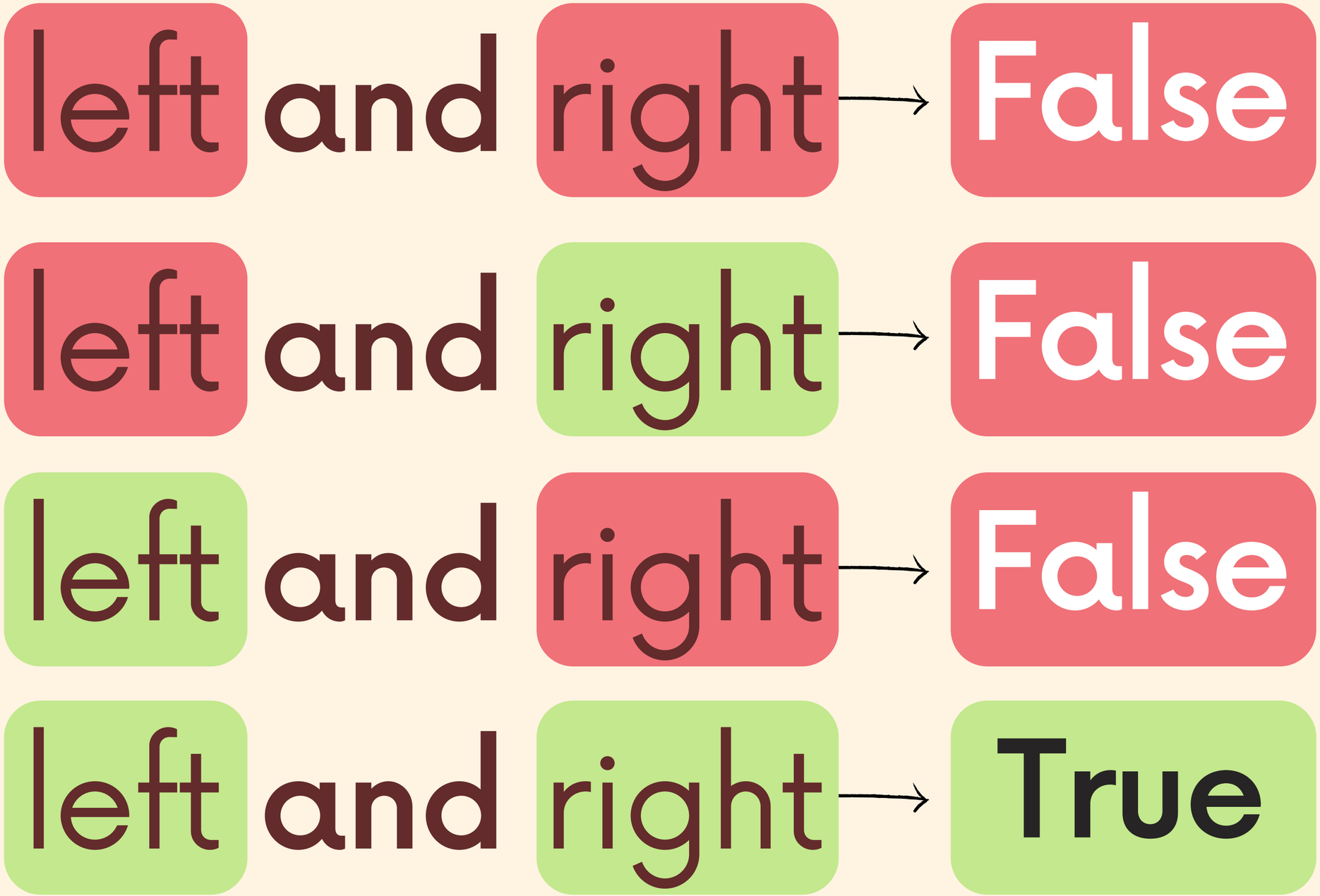
```
'a' == 'a' and 1 < 5
True
```

logical and

The **and** operator will evaluate to True only if both the left and right sides evaluate to True.

```
'a' == 'a' and 1 < 5
True
```


>>> age = 18
>>> age > 10 and age < 21
True


left and right → False
left and right → False
left and right → False
left and right → True

logical or

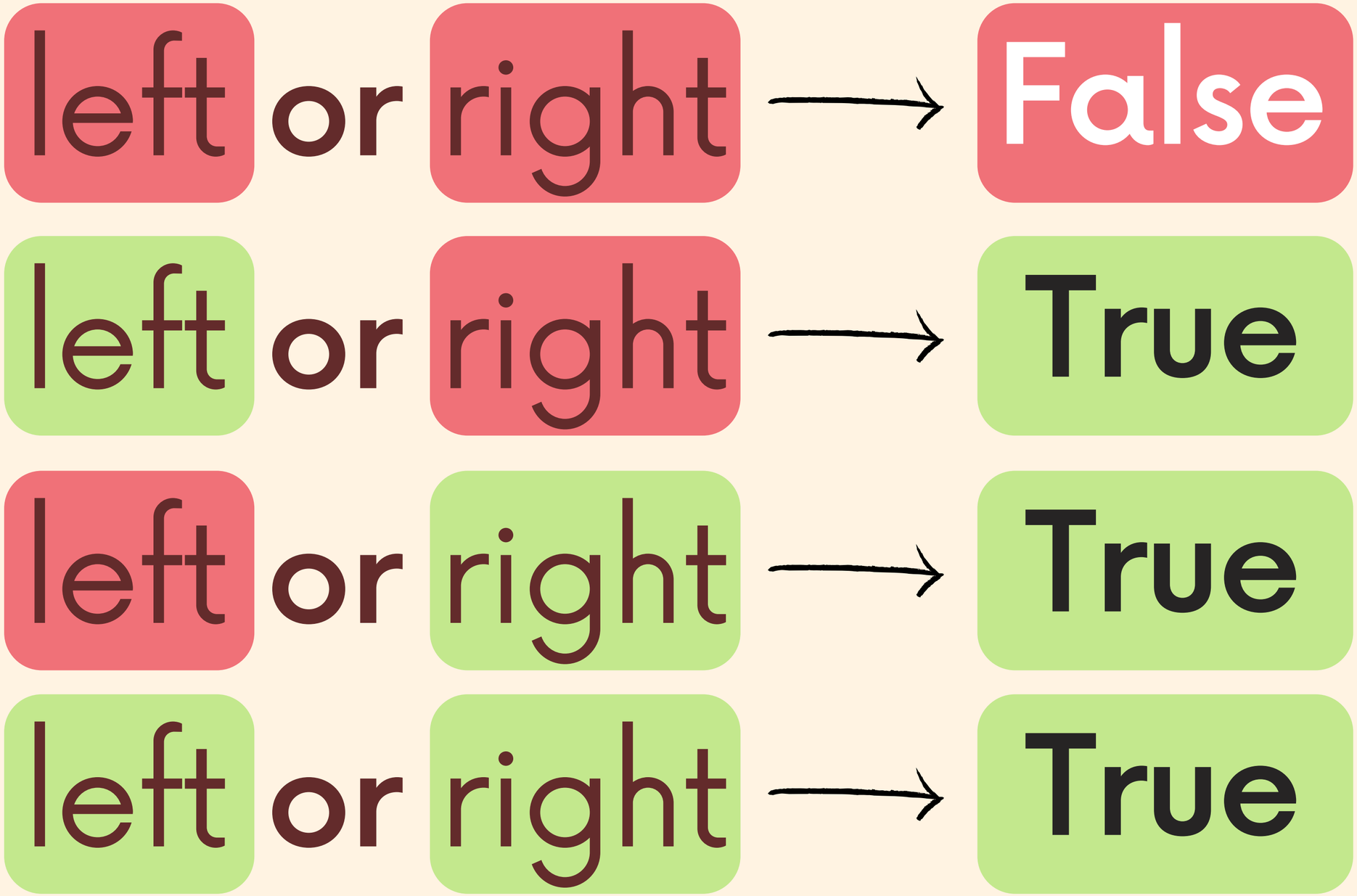
The **or** operator will evaluate to True if one or both the left or right sides evaluate to True.

```
'a' == 'b' or 1 < 5
True
```

logical or

The **or** operator will evaluate to True if one or both the left or right sides evaluate to True.

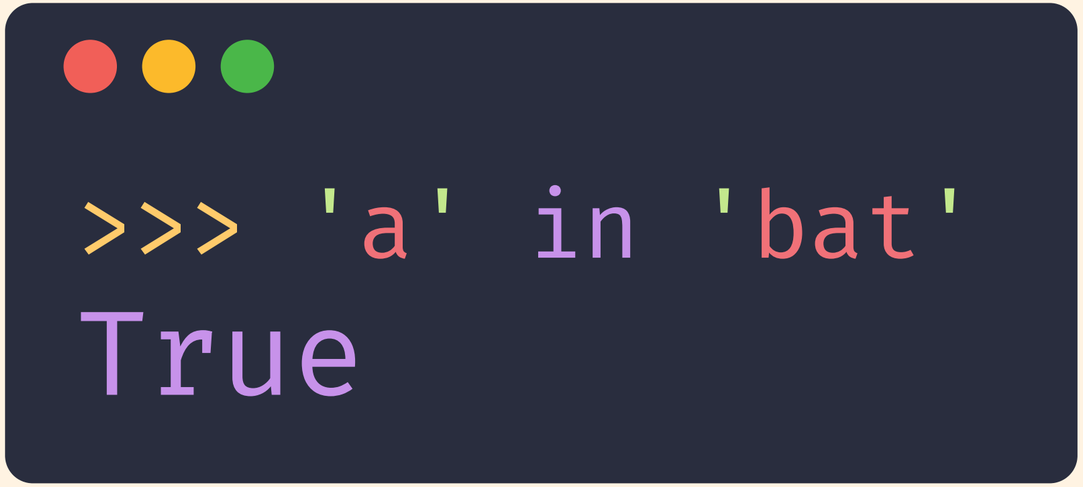
```
'a' == 'b' or 1 < 5
True
```


left or right → False
left or right → True
left or right → True
left or right → True

in

The "in" operator looks to see if an items is a member of a sequence.

Soon we'll see other sequences types!



logical **not**

The **not** operator changes True to False and False to True. It negates expressions.

```
1 < 5
True
```

```
not 1 < 5
False
```