

Conditionals

Recap: Booleans

Booleans are a data type that can take values of `True` or `False` .

Conditionals: if

Conditionals are statements that check for a condition, using the `if` statement, and then only execute a set of code if the condition evaluates as `True`.

```
In [1]: condition = True

if condition:
    print('This code executes if the condition evaluates as True.')
```

This code executes if the condition evaluates as True.

Conditional: else

After an `if`, you can use an `else` that will run if the conditional(s) above have not run.

```
In [2]: condition = True

if condition:
    print('This code executes if the condition evaluates as True.')
else:
    print('This code executes if the condition evaluates as False')
```

This code executes if the condition evaluates as True.

Conditional: elif

After an if statement, you can have any number of `elif`'s (meaning 'else if') to check other conditions.

```
In [3]: condition_1 = False
condition_2 = True

if condition_1:
    print('This code executes if condition_1 evaluates as True.')
elif condition_2:
    print('This code executes if condition_1 did not evaluate as True, but condition_2 does.')
else:
    print('This code executes if both condition_1 and condition_2 evaluate as False')
```

This code executes if condition_1 did not evaluate as True, but condition_2 does.

Conditionals With Value Comparisons

Any expression that can be evaluated as a boolean, such as value comparisons, can be used with conditionals.

```
In [6]: language = "Python"

if language == "Python":
    print("Yay!")
elif language == "Matlab":
    print("Oh no.")
else:
    print("Get yourself a programming language!")
```

Yay!

```
In [7]: # Exploring conditionals

number = 3

print('Before Conditional')

if number < 5:
    print('    If statement execution')
elif number > 5:
    print('    Elif statement execution')

print('After Conditional')
```

```
Before Conditional
    If statement execution
After Conditional
```

Clicker Question #1

What will the following code snippet print out:

```
if False:
    print("John")
elif True:
    print("Paul")
elif True:
    print("George")
else:
    print("Ringo")
```

A) John | B) Paul, George, Ringo | C) Paul | D) Paul, George | E) Ringo

Clicker Question Answer

```
In [ ]: if False:
        print("John")
        elif True:
            print("Paul")
        elif True:
            print("George")
        else:
            print("Ringo")
```

Clicker Question 2

What will the following code snippet print out:

```
if 1 + 1 == 2:  
    print("I did Math")  
elif 1/0:  
    print("I broke Math")  
else:  
    print("I didn't do math")
```

A) I did Math | B) I broke Math | C) I didn't do math | D) This code won't execute

Clicker Questions Answer

```
In [ ]: if 1 + 1 == 2:
        print("I did Math")
        elif 1/0:
            print("I broke Math")
        else:
            print("I didn't do math")
```

Clicker Question #3

What will the following code snippet print out:

```
python = "great"

if True:
    if python == "great":
        print("Yay Python!")
    else:
        print("Oh no.")
else:
    print("I'm here")
```

A) Yay Python | B) Oh no. | C) I'm here | D) This code won't execute

Clicker Question Answer

```
In [ ]: python = "great"

if True:
    if python == "great":
        print("Yay Python!")
    else:
        print("Oh no.")
else:
    print("I'm here")
```

Properties of conditionals

- All conditionals start with an `if`, can have an optional and variable number of `elif`'s and an optional `else` statement
- Conditionals can take any expression that can be evaluated as `True` or `False`.
- At most one component (`if` / `elif` / `else`) of a conditional will run
- The order of conditional blocks is always `if` then `elif(s)` then `else`
- Code is only ever executed if the condition is met