Programming Basics

Conditionals

If Statements

Vocabulary

- **Conditional** - a programming construct that allows a program to make a decision based on whether or not something is true at the time.
- The most common conditional is the “if statement”.

FOLLOW ALONG

- Type this code and run it. What happens

```
if 3 < 4:
  print("3 is less than 4")
```

- Now let’s try one that’s not true. What do you expect to see?

```
if 4 < 3:
  print("4 is less than 3")
```

- Try this, which has more than one indented line.

```
if 3 < 4:
  print("This is true")
  print("Which is why we see both of these lines")
```

DISCUSSION

- What do you expect to see here?

```python
if 4 < 3:
  print("This is true")
  print("Which is why we see both of these lines")
```

FOLLOW ALONG

Anything not indented will run whether the if statement is true or not:

```
print("This is the start of the program")
if 3 < 4:
  print ("This statement is true")
  print ("Which is why we see both of these lines")
print("This is the end of the program")
```

DISCUSSION

- What do you expect this to print?

```
print("This is the start of the program")
if 4 < 3:
  print ("This statement is true")
  print ("Which is why we see both of these lines")
print("This is the end of the program")
```

Debugging conditionals

FOLLOW ALONG

What happens when you don’t indent the line after an if?

```
if 4 < 3:
print ("This is true")
```

Error message

- `IndentationError: expected an indented block`
- This error is exactly what it sounds like - the program expects the line after the if statement to be indented. A group of indented lines (one or more) is called a “block”

FOLLOW ALONG

What happens when you don’t have a line after the if at all?

```
if 4 < 3:
```

Error message

- `SyntaxError: unexpected EOF while parsing`
- EOF stands for “End of File” - Python expects there to be a line after the if. If there isn’t one it reaches the end of the file unexpectedly.

FOLLOW ALONG

What happens when you don’t have a colon after the if:

```
if 4 < 3
```

Error message

- `SyntaxError: invalid syntax`
- In this case, Python doesn’t know what exactly you’ve done wrong, but it knows that there’s something wrong with how you wrote the program. Often this means you missed an important punctuation mark (a colon in this case).This is called a “syntax error”.

FOLLOW ALONG

What happens when the first line is indented but the second line isn't?

```
print("Start program")
if 4<3:
  print("4 is actually less than 3??")
print("I'm pretty sure 4 is not less than 3.")
print("End program")
```

Error message

- There is no error message. This code is syntactically correct.
- You’ve just written it incorrectly for what you want to happen.
- This is called “**logic** error” rather than a “syntax error”.

ACTIVITY

- See how many different ways you can break your program.
  - What error messages do you get?
    - What do they seem to mean?
  - What logic errors can you make that have no error message but are still (logically) wrong?

DISCUSSION

- What error messages did you get?
- What fixed them?
- What logic errors can you make that have no error message but are still wrong?

Equals

Vocabulary

- **Assignment** - setting the value of a variable
- **Equality** - Two values are the same

= vs ==

- In Python, you’ve seen = used for assignment, meaning we assign values to variables using =
- This means means we can’t use = for equality
- Instead, Python uses == to see if two values are equal.

DISCUSSION

- What do you expect to see here?

```
print("Start program")
if 3 == 3:
  print("3 is equal to 3")
print("End program")
```

DISCUSSION

- What do you expect to see here?

```
print("Start program")
if 3 == 4:
  print("3 is equal to 4???")
print("End program")
```

FOLLOW ALONG

Watch what happens when you use = instead of ==

```
print("Start program")
if 3 = 4:
    print("this is how equals works")
print("End program")
```

Error message

-
  ```
   File "<ipython-input-74-23fe54832588>", line 2
      if 3 = 4:
           ^
  SyntaxError: invalid syntax
  ```

- “Invalid syntax” means you’ve broken one of the ways Python expects to see things
- Python also shows you exactly where it sees a problem with the ^ symbol.

Strings

- You can also check if two strings are the same with ==

DISCUSSION

- What do you expect to see here?

```
print("Start program")
if "Sarah" == "Sarah":
  print("This is how equals works")
print("End program")
```

DISCUSSION

- What do you expect to see here?

```
print("Start program")
if "Sarah" == "someone else":
  print("This is also how equals works")
print("End program")
```

FOLLOW ALONG

Capitalization MATTERS! Watch what this outputs:

```
print("Start program")
if "Sarah" == "SARAH":
  print("These are not the same")
print("End program")
```

FOLLOW ALONG

You can use variables in if statements:

```
print("Start program")
name = "Sarah"
if name == "Sarah":
  print("The names are the same")
```

FOLLOW ALONG

Which means you can use input

```
print("Start program")
name = input("What name would you like to check?")
if name == "Sarah":
  print("The names are the same")
print("End program")
```

Not Equals

Not Equals

- Sometimes, you want to check if something is NOT the same as something else.
- != in Python can be read as “not equals”

DISCUSSION

- What do you expect to see here?

```
print ("Start program")
if "A" != "B":
  print ("A does not equal B")
print ("End program")
```

DISCUSSION

- What do you expect to see here?

```
print ("Start program")
if "A" != "A":
  print ("A does not equal A")
print ("End program")
```

Else

DISCUSSION

What do you expect this to print?

```
print("Start program")
if "Sarah" == "not Sarah":
  print("The names are the same")
else:
  print("The names are not the same")
print("End program")
```

Else if

- Sometimes you have multiple conditions you want to check.
- You can use the keyword `elif` which is short for “else if”.
- Note that you should only use elif when the condition for the elif is dependent on the previous condition.

DISCUSSION

What do you expect this to print?

```
print("start program")
guessed_name = "Ms. Judd"
if guessed_name == "Sarah":
  print ("You guessed correctly")
elif guessed_name == "Ms. Judd":
  print ("I also go by that name")
else:
  print("That's not my name")
print("end program")
```

DISCUSSION

What do you expect this to print?

```
print("start program")
guessed_name = "Elizabeth"
if guessed_name == "Sarah":
  print ("You guessed correctly")
elif guessed_name == "Ms. Judd":
  print ("I also go by that name")
else:
  print("That's not my name")
print("end program")
```

DISCUSSION

What do you expect this to print?

```
print("start program")
guessed_name = "Sarah"
if guessed_name == "Sarah":
  print ("You guessed correctly")
elif guessed_name == "Ms. Judd":
  print ("I also go by that name")
else:
  print("That's not my name")
print("end program")
```