APS106

UNIVERSITY OF TORONTO

Rock, paper, scissors, lizard, Spock.

Design Problem 2

While waiting for class to start:

Download and open the Jupyter Notebook (.ipynb) for Lecture 2.2.1

You may also use this lecture's JupyterHub link instead (although opening it locally is encouraged).

Upcoming:

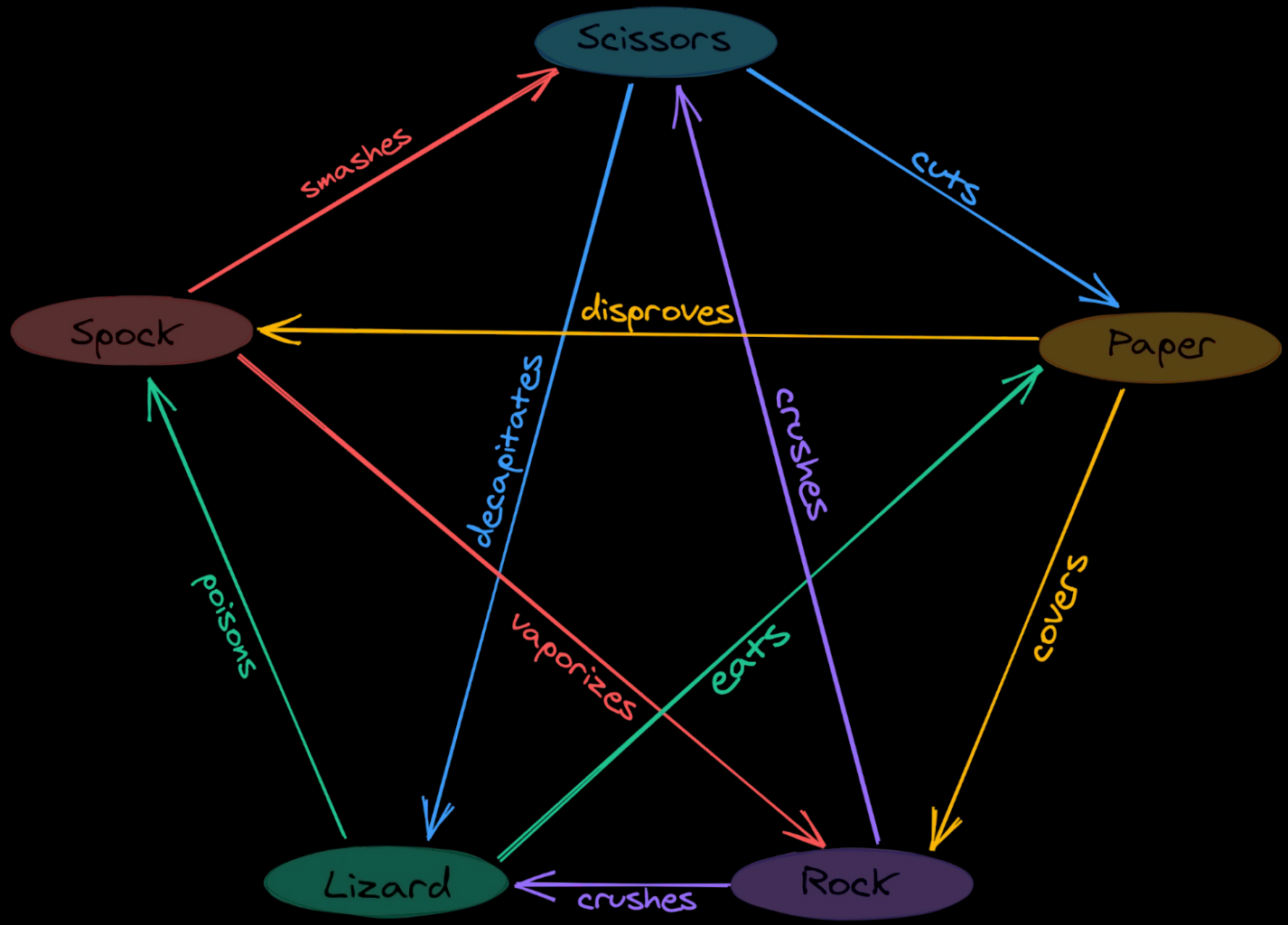
- Reflection 2 released Friday @ 11 AM
- Lab 3 released Friday @ 12 PM
- Lab 2 **deadline** this Friday @ 11 PM
- PRA (Lab) on Friday @ 2PM this week (ONLINE)

if nothing else, write #cleancode

Our problem background starts in a famous TV show apartment...

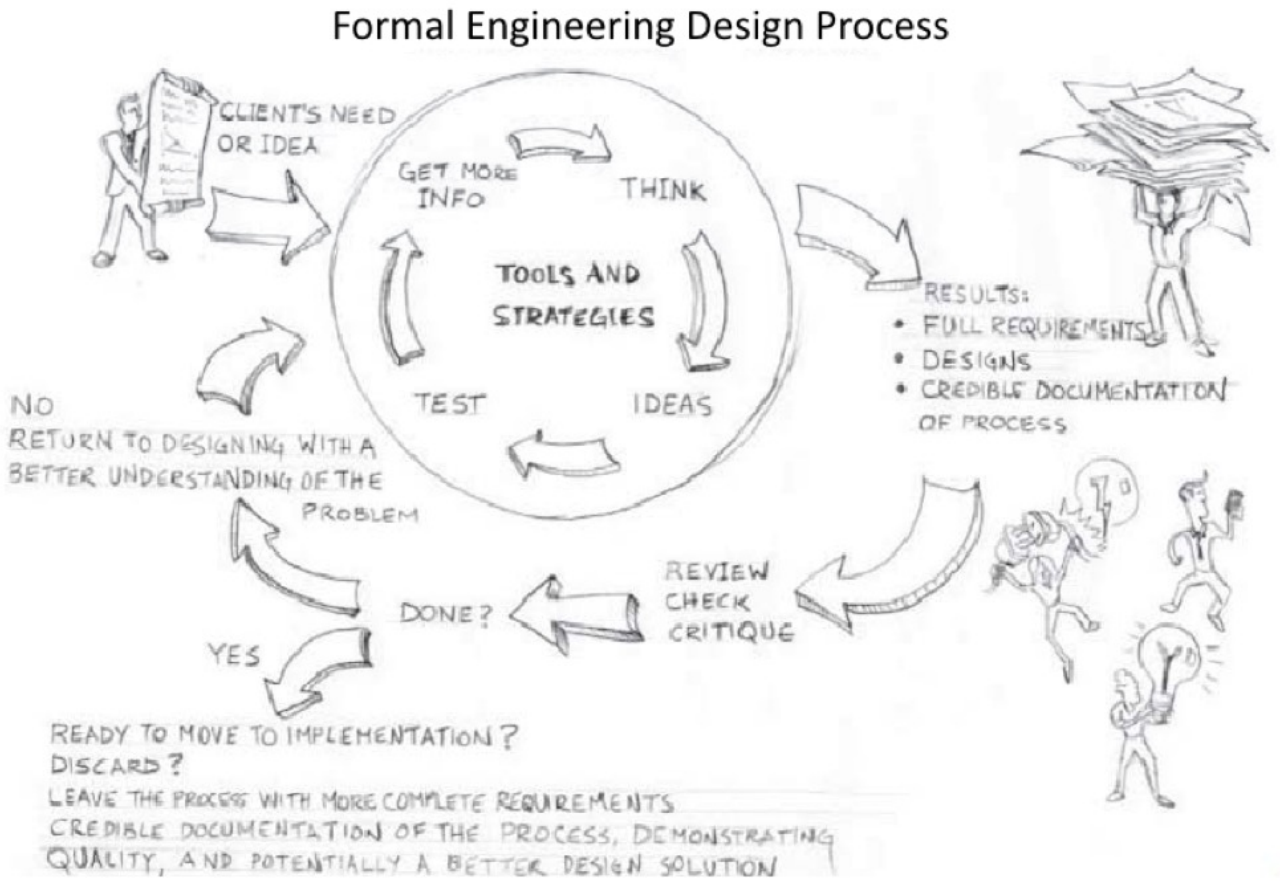
Background

- Use conditional statements and imported modules to create a basic computer game!
- **Goal:** Write a Python program that using conditional statements and user input to play Rock-Paper-Scissors-Lizard-Spock against a computer program with random choice selection.



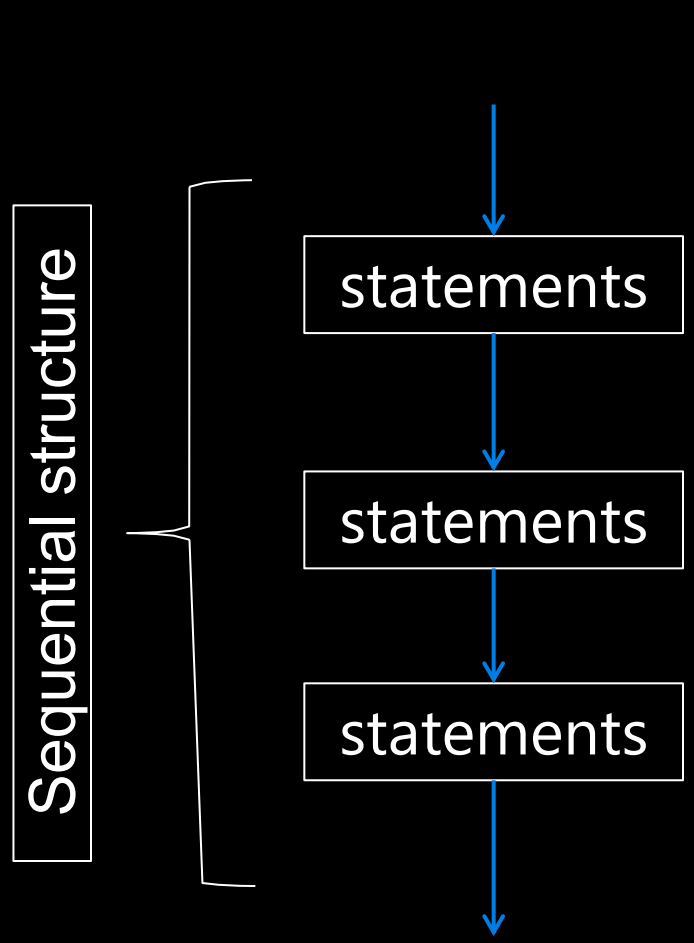
Engineering Design Process

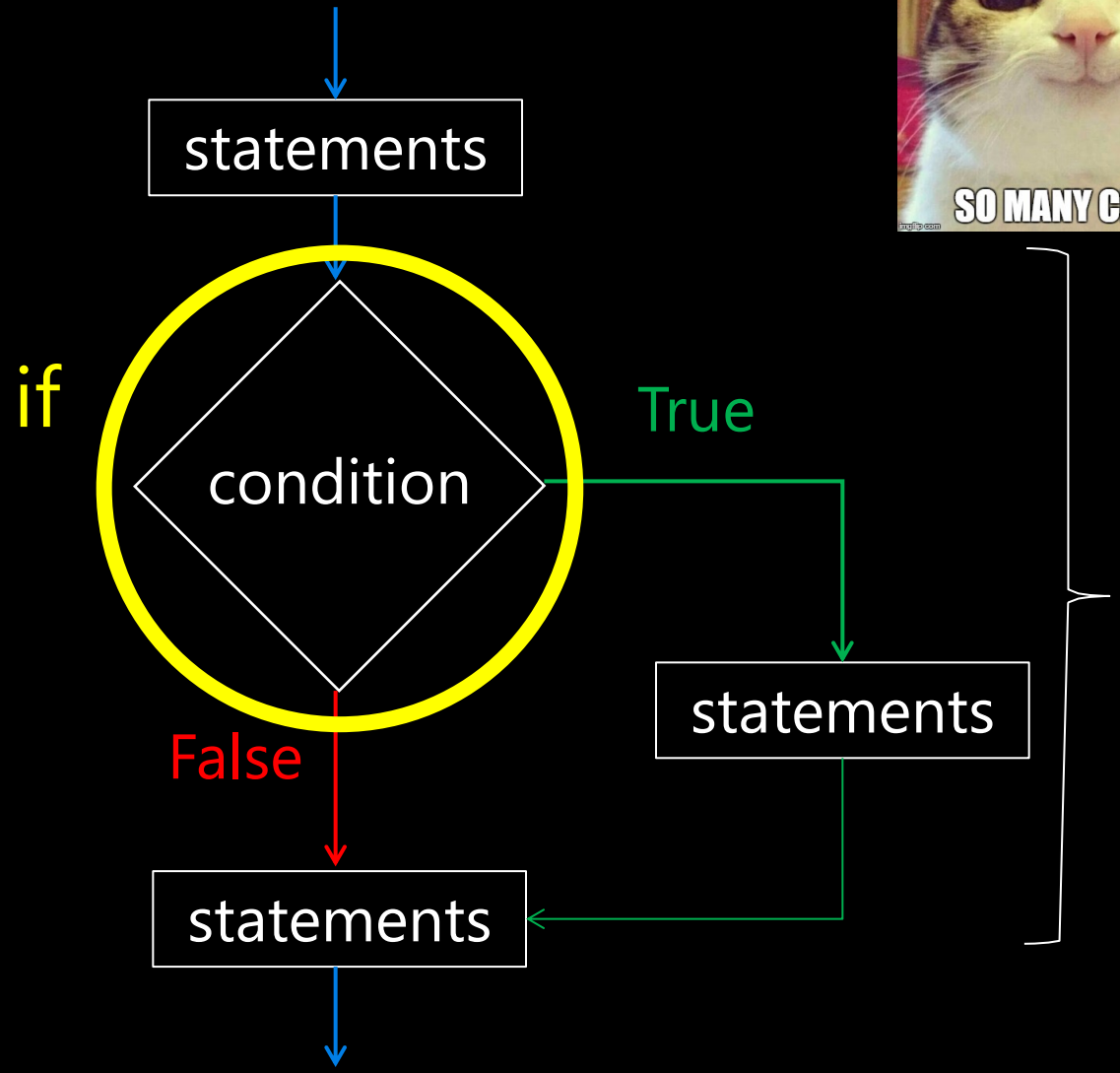
- Learn to define the problem.
- Practice defining test cases.
- Develop an algorithm plan (i.e. a workflow!).
- Program your solution and debugging.



Some friendly reminders…

RECAP: Making Choices





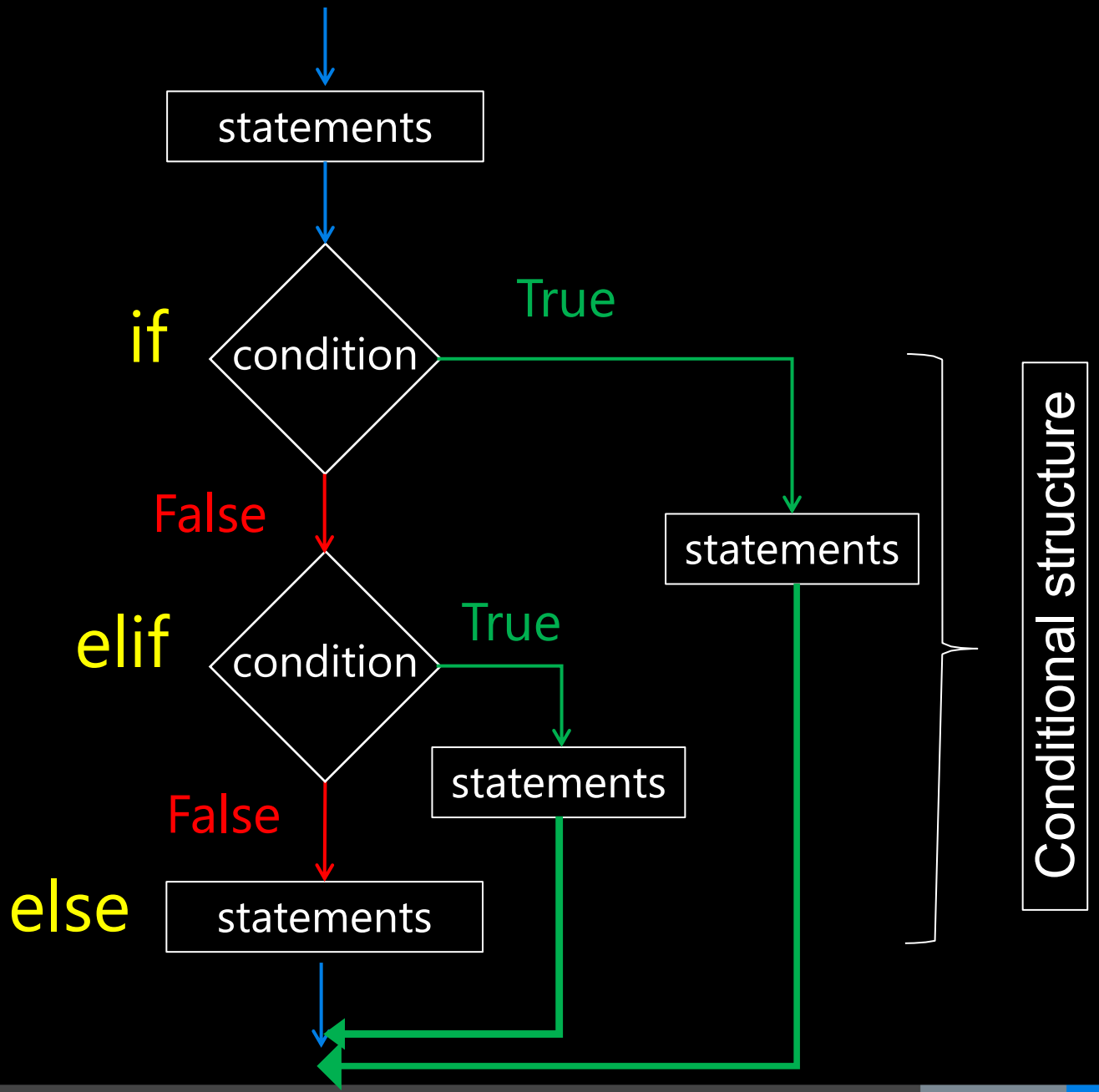
Adding the elif (else if) statement

- The most general form of the if conditional statement is:

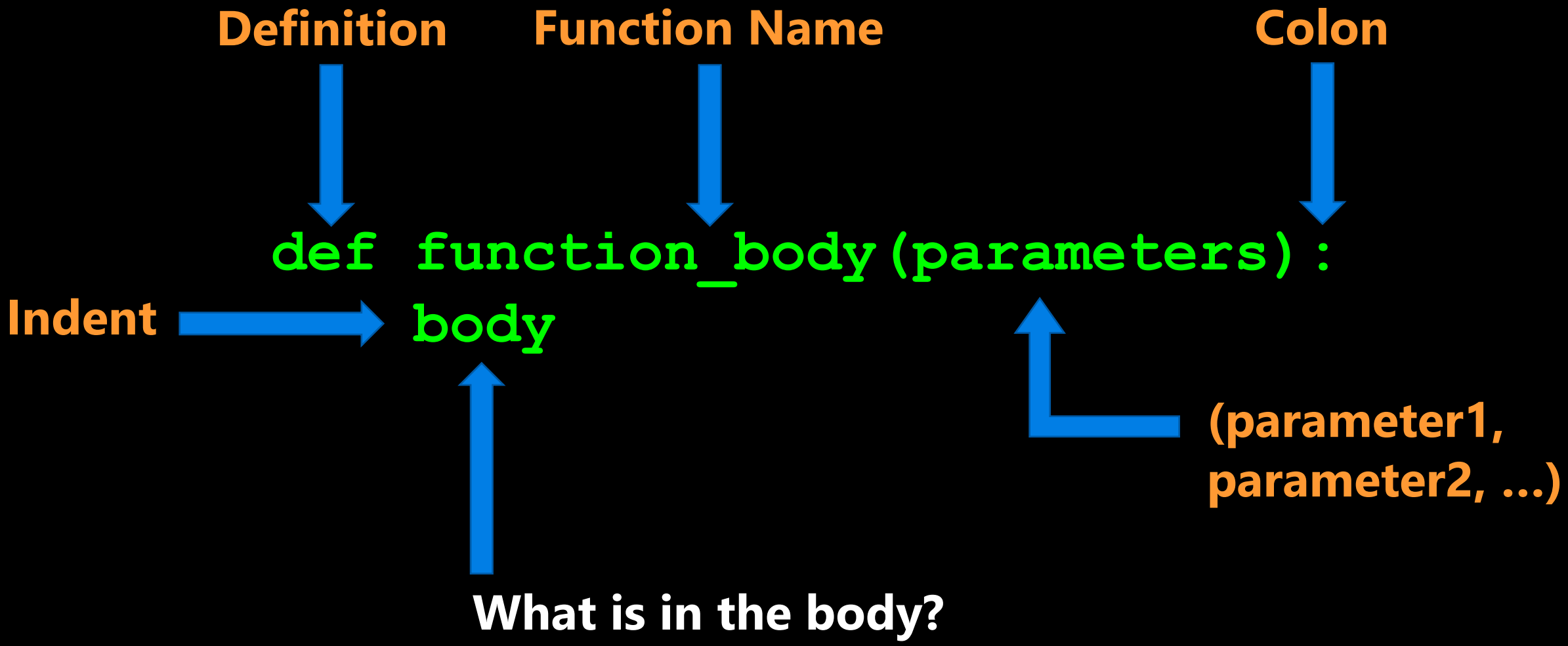
```
if condition1:
    body1
elif condition2:
    body2
elif conditionN:
    bodyN
else:
    other_body
```

- Note the colons (:) and the indents!
- ONLY 1 body will be executed.
  - if statement is True, execute body1, exits if structure
  - if statement is False, continue to elif statement
  - elif statement is True, execute elif body, exits if structure
  - elif statement is False, continue to next elif statement
  - All if's and elif's are False, execute else statement

Making Choices



Function Definitions



Function Definitions

```
def function_body(parameters):
```

1. """DOCSTRING""" (optional)

2. Code the does the thing

3. return [expression]
   The return statement is options and if it is not included, it's the same as writing return None

Calling Functions

- The general form of a function call:

```
function_name(arguments)
```

- Terminology
  - *argument*: a value given to a function.
  - *pass*: to provide an argument to a function.
  - *call*: ask Python to execute a function (by name).
  - *return*: give a value back to where the function was called from.

In **Python** names of variables and functions use low case and underscores.

```
function_name
Function_Name
FunctionName
```

Input

- Python has a built-in function named **input** for reading text from the user.
- The general form of a **input** function call:

```
input(argument)
```

- The argument is the text you want displayed to the user.
  - *"What is your name?"*
- The value returned by the **input**function is always a string.

String Comparisons

- Boolean comparisons can also be applied to strings, whether single characters or sets of characters
- Compare two strings by their **dictionary order**, comparing letter by letter

Description	Operator	Example	Result of example
equality	==	'cat' == 'cat'	True
inequality	!=	'cat' != 'Cat'	True
less than	<	'A' < 'a'	True
greater than	>	'a' > 'A'	True
less than or equal	<=	'a' <= 'a'	True
greater than or equal	>=	'a' >= 'A'	True

Importing Functions and Modules

- The general for of an import statement is:
  - `import module_name`
- To access a function within a module:
  - `module_name.function_name`

APS106

UNIVERSITY OF TORONTO

Rock, paper, scissors, lizard, Spock.

Design Problem 2

if nothing else, write #cleancode