Beginning Python: Lesson 2

created by David Bau, updated by Tramore Coderdojo

In this lesson we will build on what we learned in lesson 1 to make the game hangman. **You will need to have Python version 3 installed on your computer** to complete this lesson. If you have an older version of Python, please download and install version 3. If you're unsure, ask a mentor for help.

Goals for Today

- Create your first Python Program
- Use Wings IDE to run a Python Program
- Create Hangman Game!

1: Before You Start...

Make sure you have installed Python and Wing IDE 101. This is covered in last weeks lesson.

2: Your first program

A program is just a file with lines of code in it. We can create programs in Wing IDE.

i. Start **Wing IDE 101** on your laptop.
ii. Click on the New button and enter the following code into the Editor:

```
#!/usr/bin/env python

print('time to play hangman')
```

The first line tells the computer which what language the program is written in: python. Everything after that is the code.Enter the name "hangman.py" and save the file on your desktop.

Now run your program by clicking on the Run button. The program will print "time to play

hangman" in the console.
Congratulations - you've just written your first Python Program.

3: Programs are Script.

A program is just a script of instructions that a computer reads as quickly as it can. If we put a lot of lines of code into a program, the computer will do them all, one after the other.Try typing these extra lines into your program. Don't forget to save the file again.

```
#!/usr/bin/env python
print('time to play hangman')
secret = 'crocodile'
for letter in secret:
            print('_', end = ' ')
```

the `end = ' '` part in `print(letter, end = ' ')` tells the computer to put a space after the letter. This makes all the letters appear on the same line.
Now run the program again by clicking the "run" button.
Check out the console to see what happened.

4: Using "if" to Choose

In our hangman game, we should show where any guessed letters are. To decide whether to print a line or a letter, we will need to use "if" and "else".
Now change your program to the following:

```
#!/usr/bin/env python
print('time to play hangman')
secret = 'crocodile'
guesses = 'aeiou'
for letter in secret:
        if letter in guesses:
            print(letter, end = ' ')
        else:
            print('_', end = ' ')
```

Don't forget to line everything up, and remember to save it.
What happens when you run it?
The line "if something in something:" makes a choice. If the letter is in our guesses, it prints the letter. Otherwise it prints a little line ("else:" is how you say "otherwise" in python). Since the whole thing is under the "for something in something", this choice is repeated for every letter.

5: Getting input from the User.

The Hangman game is no good unless you can guess the missing letters. To let the player guess numbers, we will use a function called "raw_input()". We can get guesses from the player like this:

```
answer = input( 'my question' )
```

Change your program to look like this.

```
#!/usr/bin/env python
print('time to play hangman')
secret = 'crocodile'
guesses = 'aeiou'
guess = input('guess a letter: ')
guesses = guesses + guess
for letter in secret:
        if letter in guesses:
            print(letter, end = ' ')
        else:
            print('_', end = ' ')
```

The "guesses = guesses + guess" line add the guess to the string of guesses. If the string of guesses was "aeiou" and the new guess is "c", then the string of guesses will become "aeiouc".
Run it to check how it works...

6: Using while to Repeat

We need to let the player take more than one turn. The "while" command can repeat our program until the player is out of turns. This is like the repeat block we used in Scratch.

```
#!/usr/bin/env python
print('time to play hangman')
secret = 'crocodile'
guesses = 'aeiou'
turns = 5

while turns > 0:
    for letter in secret:
        if letter in guesses:
            print(letter, end = ' ')
        else:
            print('_', end = ' ')
    print();
    guess = input('guess a letter: ')
    guesses = guesses + guess
    turns = turns - 1
```

You need to indent everything under the "while" command to make this work. So you will need to add some spaces in front of most of your program.
Let's also move the guessing after the hint instead of before.
The command 'turns = turns - 1' means subtract one from "turns," so if it used to be 5, it will be 4. Then the next time around it will be 3 and so on. When turns is finally zero, the "while" command will stop repeating. Try running your program. Does it work?

7: Improving the Game

We can already play our game. Now we need to fix it up so that it is fun.
1. The player should win right away when there are no missing letters.
2. The player should only lose a turn on a wrong guess.
3. When the player loses, the game should tell the secret.
Put in the following code:

```
#!/usr/bin/env python
print('time to play hangman')
secret = 'crocodile'
guesses = 'aeiou'
turns = 5

while turns > 0:
    missed = 0
    for letter in secret:
        if letter in guesses:
            print(letter, end = ' ')
        else:
            print('_', end = ' ')
            missed = missed + 1
    print();
    if missed == 0:
        print('YOU WIN!!!!!')
        break
    guess = input('guess a letter: ')
    guesses = guesses + guess
    if guess not in secret:
        turns = turns - 1
        print('Wrong guess')
        if turns == 0:
            print('Game Over, the answer is ', secret)
```