PROBLEM!

- There is so much going on in my main function; or
- I keep repeating myself over and over

~~~~python
from random import randint

low_number = 1
high_number = 100

max_guesses = 10
current_guess = 0

guess = 0
answer = randint(low_number, high_number)

while guess != answer and max_guesses - current_guess > 0:

    guess = int(input(f"Guess a number between {low_number} and {high_number}. You have {max_guesses-current_guess} remaining: "))

    if guess < low_number or guess > high_number:
        continue

    if guess < answer:
        print(f"{guess} is too low")
    elif:
        print(f"{guess} is too high")

    current_guess = current_guess + 1
print(f"That's good, {guess} was the right number!!!!")
    ```
~~~~

WOULDN'T IT BE GOOD IF WE COULD MOVE STUFF OUT OF THE LOOP?

Well, we can. We can also do this to remove repitition.

HOW?

```python
from random import randint

min_value = 1
max_value = 100
answer = randint(min_value, max_value)
guess = -1
moves = 0

def get_guess(min_value, max_value):
    guess = int(
        input(
            f'''Guess a number between {min_value} and
            {max_value} '''
        )
    )
    return guess

while answer != guess:
    guess = guess(min_value, max_value)

    if guess < min_value or guess > max_value:
        continue

    moves = moves + 1
    if guess < answer:
        return "your guess was too low!"
    elif guess > answer:
        return "your guess was too high!"


print(f"{guess} was the right answer. You won in {moves} moves.")
```

REFACTORING

```python
from random import randint

min_value = 1
max_value = 100
answer = randint(min_value, max_value)
guess = -1
moves = 0

def get_guess(min_value, max_value):
    guess = int(
        input(
            f'''Guess a number between {min_value} and
            {max_value} '''
        )
    )
    return guess

while answer != guess:
    guess = guess(min_value, max_value)

    if guess < min_value or guess > max_value:
        continue

    moves = moves + 1
    if guess < answer:
        return "your guess was too low!"
    elif guess > answer:
        return "your guess was too high!"


print(f"{guess} was the right answer. You won in {moves} moves.")
```

GETTING RID OF BAD CODE SMELLS

```python
def get_hint(guess, answer):
    if guess < answer:
        return "your guess was too low!"
    elif guess > answer:
        return "your guess was too high!"

while answer != guess:
    guess = get_guess(min_value, max_value)

    if guess < min_value or guess > max_value:
        continue

    moves = moves + 1
    print(get_hint(guess, answer))
```

FIXING MORE SMELLS!

```python
while answer != guess:
    guess = get_guess(min_value, max_value)

    if guess < min_value or guess > max_value:
        continue

    moves = moves + 1
    print(get_hint(guess, answer))
```

LAST FUNCTION

```python
from random import randint

min_value = 1
max_value = 100
answer = randint(min_value, max_value)
guess = -1
moves = 0

def get_guess(min_value, max_value):
    guess = -1
    while guess < min_value or guess > max_value:
        guess = int(
            input(
                f"Guess a number between {min_value} and {max_value} "
            )
        )
    return guess

def hint(guess, answer):
    if guess < answer:
        return "your guess was too low!"
    elif guess > answer:
        return "your guess was too high!"

while answer != guess:
    guess = get_guess(min_value, max_value)
    moves = moves + 1
    print(hint(guess, answer))


print(f"{guess} was the right answer. You won in {moves} moves.")
```

ONE LAST ODDITY

```python
from random import randint

min_value = 1
max_value = 100
answer = randint(min_value, max_value)
guess = -1
moves = 0

def get_guess(min_value, max_value):
    guess = -1
    while guess < min_value or guess > max_value:
        guess = int(
            input(
                f"Guess a number between {min_value} and {max_value} "
            )
        )
    return guess

def hint(guess, answer):
    if guess < answer:
        return "your guess was too low!"
    elif guess > answer:
        return "your guess was too high!"

while answer != guess:
    guess = get_guess(min_value, max_value)
    moves = moves + 1
    print(hint(guess, answer))


print(f"{guess} was the right answer. You won in {moves} moves.")
```

OKAY, SO PARTS OF A FUNCTION

1. Function signiture: must be unique - also don't name it something that python already has
2. Function body: is indented, and runs procedurally from top to bottom
3. Function call: this is where the function is called

```python
def greet_user():
    print("Hello!")
greet_user()
```

CALLING FUNCTIONS

```python
def greet_user():
    print("Hello!")
greet_user()
greet_user()
greet_user()
```

```
> "Hello"
> "Hello"
> "Hello"
```

PASSING ARGUMENTS TO FUNCTIONS

- Generally speaking we will want to pass something from our main thread to the function
- Function body: is indented, and runs procedurally from top to bottom

```python
def my_addition(a, b):
    print(a + b)
my_addition(1, 2)
my_addition(4, 1)
my_addition(5, -1)
```

FUNCTIONS SHOULD RETURN

- Generally speak, our functions should return some value to the main thread
- Sometimes we can't, and that's okay, but we should always ask ourselves why aren't we?

```python
def my_addition(a, b):
    return a + b
print(my_addition(1, 2))
print(my_addition(4, 1))
print(my_addition(5, -1))
```

PYTHON CAN RETURN MULTIPLE VARIABLES

- Sometimes your function wants to modify multiple variables at the same time

```python
def get_guess(min_value, max_value, moves):
    guess = int(
        input(
            f'''Guess a number between {min_value} and
            {max_value} '''
        )
    )
    moves = moves + 1
    return guess, moves


guess, moves = get_guess(1, 10, guess)
```

Speaker notes