

# More about functions



FEB 23RD

# Keyword arguments



- You can avoid matching by position by using *keyword arguments* in the function call.
- **Example:** `manyRandomWalks(numRepetitions = 200, n = 20)`
- Here `numRepetitions` and `n` are function parameters.
- Since the actual parameters are explicitly being provided values in the function call, the matching of arguments to parameters is no longer positional.
- The above function call is identical to the call `manyRandomWalks(n = 20, numRepetitions = 200)`

# Keyword parameters



- There is a way to define *default* values of parameters.
- **Example:** `def manyRandomWalks(n, numRepetitions = 100)`
- This function can now be called with one or two arguments and in different styles.
- **Examples:** Try these out
  - `manyRandomWalks(10)`  
(The default value of 100 is used for `numRepetitions`; 10 is used for `n`)
  - `manyRandomWalks(40, 150)`  
(40 is used for `n`, 150 for `numRepetitions`)

# Another example



```
def test(x = 3, y = 100, z = 200):  
    return x - y + z
```

## Examples of function calls:

1. `test(10)` (10 is used for `x`; default values 100 for `y` and 200 for `z`)
2. `test(10, 20)` (10 is used for `x`, 20 for `y`; default value 200 for `z`)
3. `test(z = 35)` (default values 3 for `x`, 100 for `y`; 35 for `z`)
4. `test(10, z = 35)` (10 for `x`, default value 100 for `y`, 35 for `z`)
5. `test(z = 50, 10, 12)` (Error: positional arguments come first, then keyword arguments)

# Things that functions return



- Functions don't have to explicitly return values. For example:

```
def printGreeting(name):  
    print "Hello", name, "how are you?"
```

- How would you call such a function?

## **Example:**

```
printGreeting("Michelle")
```

- What would happen if you executed?

```
x = printGreeting("Michelle")
```

# The object `None`



- It is used by Python to represent the absence of a value.
- It has a type called `NoneType` and `None` is the only object of this type.
- `None` has a boolean value that is `False`.

# Functions practice problem 1



- Write a function called `search` that reads a sequence of words (strings) one per line, looking for the word “hello.” The function should assume that the sequence will be terminated by the empty string.
- **Enhancements:**
  1. Make the function have a keyword parameter that represents the word it is searching for. Have the default value of this be “hello.”
  2. Make the function have an additional parameter that represents the number of words it is willing to read while waiting for the word it is looking for.

# Functions Practice Problem 2



- Write a function that simulates the roll of two 6-sided dice 100 times and returns the number of times 4 shows up as the sum of the outcomes on the two dice.
- **Enhancements:**
  1. Make the function take the number of times it needs to roll the dice as a parameter, with 100 being the default value.
  2. Make the function take the number of sides of the die as a parameter, with 6 being the default value.
  3. Make the function take the number of dice it needs to roll as a parameter, with 2 being the default value.