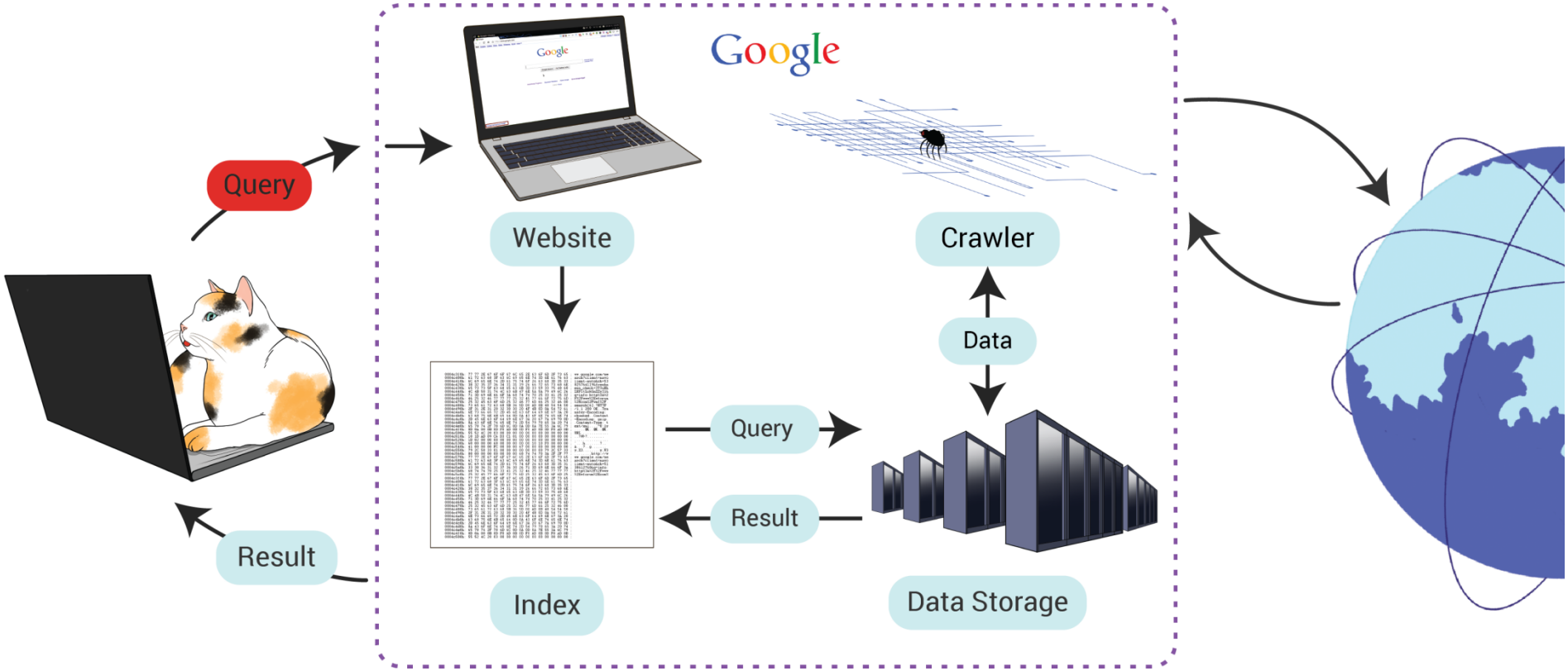
Announcements

- Relationship to section-001 and 003
- In-class quiz

Today

- Search engine and basic python wrap-up
- Arrays, boolean conditions, and functions
- Python →ISA → Numbers ☺ → Machine

Search Engine



Search part

```
search_word = "the"
file1_name = "Ode on the death of a favorite cat"
file1_word1 = "twas"
file1_word2 = "on"
file1_word3 = "a"
file1_word4 = "lofty"
file1_word5 = "vases"
file1_word6 = "side"
file2_name = "Elegy written in a country churchyard"
file2_word1 = "the"
file2_word2 = "curfew"
file2_word3 = "tolls"
file2_word4 = "the"
file2_word5 = "knell"
file2_word6 = "of"
file2_word7 = "parting"
file2_word8 = "day"
```

```
if(search_word == file1_word1):
   print(file1_name)
if(search_word == file1_word2):
   print(file1_name)
if(search_word == file1_word3):
   print(file1_name)
if(search_word == file1_word4):
   print(file1_name)
if(search_word == file1_word5):
   print(file1_name)
if(search_word == file1_word6):
   print(file1_name)
if(search_word == file2_word1):
   print(file2_name)
if(search_word == file2_word2):
   print(file2_name)
if(search_word == file2_word3):
   print(file2_name)
if(search_word == file2_word4):
   print(file2_name)
if(search_word == file2_word5):
   print(file2_name)
if(search_word == file2_word6):
   print(file2_name)
if(search_word == file2_word7):
   print(file2_name)
```

Arrays

variable_name = [*expression*, *expression*, ...]
variable_name[j] : extracts j'th element in array

```
some_nums = [4,6,99,-1]
x = some_nums[0]
some_nums[1] = 4
array1 = [1, 10, 11]
array2 = [4, 15, 40]
array_new = array1 + array2
```

Strings arrays

```
s = “hello world”
y = s[0]
```

Array of arrays

```
cool_array = [ [1, 4, 11], [7, 10, 45], [8, 100, 14, 78, 101] ]
cool_array[0][0] = ?
cool_array[1][2] = ?
cool_array[2]
```

Search revisited

```
file1_name = "Ode on the death of a favorite cat"
file1_number_of_words = 6
file1_words = ["twas","on","a","lofty","vases","side"]
search_word = "the"
counter = 0
while(counter < file1_number_of_words):
   if(file1_words[counter] == search_word):
      print(file1_name)
   counter = counter + 1
```

Search re-revisited

```
number_of_files = 2
file_names = ["Ode on the death of a favorite cat",
              "Elegy in a county churchyard"]
file_word_counts = [6,8]
files = [["twas","on","a","lofty","vases","side"],
    ["the","curfew","tolls","the","knell","of","parting","day"]]
search_word = "the"
file_counter = 0
while(file_counter < number_of_files):
   word_counter = 0
   while(word_counter < file_word_counts[file_counter]):
      if(files[file_counter][word_counter] == search_word):
         print(file_names[file_counter])
      word_counter = word_counter + 1
   file_counter = file_counter + 1
```

Functions

```
def function_name(variable_name):
    [lines of code that comprise the function]
    return expression
```

```
def mogrify(x):
    y = x+1
    return 5*y+9*x*x
z = mogrify(2)
print(z)
```

Functions

```
def foobar(x, a):
    y = x+a
    return 5*y+9*x*x+a
z = mogrify(2,10)
print(z)
```

Variable scope and naming

- Read 3.5 – we will discuss on Monday and wrapup

Today

- Arrays & Functions
- They are there for making programming easier

09/18

2 minute quiz start

Announcements:
1) This course covers the same concepts as the other sections, just in a different way
2) You can rate lectures on the course website.
3) Historic grades from last semester were shown, grade distribution is expected to be similar

Today:
Search engine and basic python wrapup
Arrays, booleans, functions
Python -> ISA -> Numbers -> Machine

Search engine example
- high level previously introduced in chapter one
- website -> index -> data -> result -> user

Search part:
everything is stored in files, files have different names. Words within the files are stored in indiv variables
What a very basic search engine does is run through the list of words and look for matches
But this is AMAZINGLY inefficient and very unproductive

Arrays can help optimize this a bit!
Arrays are structures that hold multiple other entities in them
var_name = [expression, epression...]
To get something out of the array:
var_name[i] gets the ith element. Arrays are 0 indexed!!!

```
nums = [2,5,3]
x = nums[1] # x = 5
arr1 = [1, 2, 3]
arr2  = [4, 5, 6]
arrFinal = arr1 + arr2
```

The original arr1 and arr2 are unmodified.

If you try to access an array index that doesn't exist, you get an error.
Depending on the language, you may get a NULL value, a runtime error, ArrayIndexOutOfBounds...

Arrays are very complicated in terms of how the compiler sees them!

String arrays!
```
s = "hello world!"
y = s[0] #value is "h"
```

Note that spaces also count as characters in the array!!

Array of arrays (arrayception)
arrays = [[1, 2, 3], [5, 6, 7], [8, 9, 0]]

```
    arrays[0][0] = 1 #0th element of the 0th array
    arrays[2] = [ 8, 9, 0] #the 2nd array
```

So. Search, revisited:
Instead of storing the words in indiv variables, we simply place them all in an array.
This makes for clearer code.

You could also put each file array in an array of file words
files = [["i'm", "a", "file"], ["i'm", "another", "file"]]

Functions

```
def func_name(parameter):
    # code code code
    return value


def transmobulate(x):
    y = x +1
    return y

x = transmobulate(2)
print(x) # prints 3

def transmogrify(x, a):
    y = x + a
    return y
```

What happens if you modify a or x inside the function?
Name/scoping issues. Discussed in 3.5.

The whole point is to make these more readable.