Data Types

All programs are composed of two items: Data and Operations on that Data. Because, at their heart, computers are simple devices, they can only represent very simple pieces of information. All complex information must be built up from these basic **Data Types**. The data types can roughly be described as: numbers, booleans, characters, arrays, and structures. Some languages like Ruby replace characters with "strings". Object oriented languages, such as C++ and Java replace "structures" with "objects".

All programs involve storing and manipulating data.

Luckily (???) the computer only knows about a few types of data. These include, numbers, true/false values, characters (a,b,c,1,2,3,etc), lists of data, and complex "Structures" of data, which build up **new** data types by combining the other data types.

Here is a brief summary of the available data types:

- Numbers, (e.g., 7, 3.14)
- Booleans (true or false)
- Strings (a list of characters ("hello", "jim"))
- Arrays (a list of data)
- Objects (a collection of named data referring to a single entity (also contains functions on that data))

All computer programs, from brain scanners to video games to music players, use these same basic data types to represent all possible information.

It should be noted that computer programs often **approximate** information, that is they represent information **inaccurately** or perhaps it would be better to say **imprecisely**. An example of this is the value of pi. For most of us, pi is roughly 3.1415 and this value works just fine. For engineers pi is 3.14159265358979323846. For mathematicians and scientists, pi might be 3.14159265358979323846264338327950 29. Thus it can be said that a computer deals in abstractions and approximations.

Below are some more examples and specifics for the various data types.

- Numbers

```
students_listening = 112      # an "integer" (whole) number
average_number_of_students_in_class = 89.5 # a "double" (a real)
number
```

- Booleans (true or false)

```
utah_won_the_game = true
```

Warning: The two Booleans values are **true** and **false**, (not the strings 'true' and 'false' but **keywords** true and false. Further, while some languages allow you to use 1 and 0 for true and false, every time you write a program and need to assign the value's true or false, you should use the keywords **true** and **false**, not the shortcut 1,0.

- strings ('a', 'b', ... 'z', '1', '2', ... '9', '!', '^', etc)
  ```
  middle_initial = 'j'
  ```

- Arrays (a lists of data (of the SAME TYPE!))
  ```
  student_grades = [97, 78, 88, 93, 89]
  ```

You can find more info on arrays under the chapter explicitly dealing with arrays.

- Structures are a way to create more complex "Data Types" than the basics. They basically allow you to "build up" bigger and more interesting collections of data by **naming** sub-parts of information.

Below, we create a "student structure" containing, an integer part (the age), a string part (the name), and a boolean part (the fact that tuition has been paid or not).

```
student.name = 'jim'          # array of characters, or string, part
student.age  = 27             # an integer part
student.paid_tuition = false  # a boolean part
```

For more information on Structures, please read the chapter devoted to the subject.

Data Type Worksheet Name_____

Choose the data types for the variable based on the statements below:

1. days_of_homework = 77
   a) string
   b) number
   c) array
   d) Boolean

2. types_of_homework = “math”
   a) array
   b) Boolean
   c) String
   d) Number

3. days_of_homework = “77”
   a) Boolean
   b) number
   c) string
   d) array

4. do_homework = true
   a) number
   b) array
   c) string
   d) Boolean

5. do_homework = “true”
   a) Boolean
   b) String
   c) Array
   d) Number

6. Create your own Boolean variable, and assign it a value:

   _____

7. Create your own number variable, and assign it a value:

   _____

8. Describe the difference between the number 1 and a string “1”