Learning Objectives - Variables

- **Understand the rules of naming a variable**
- **Assign/overwrite a value to a variable**
- **Understand four basic data types: strings, boolean, ints, and floats**

Variables

What is a Variable?

In computer science, we often need to use data. **Variables** are used to store a value.

Each variable in Java has:

1. a data type
2. a name
3. a value

We will discuss each of these parts over the rest of this reading assignment.

Three actions for variables

There are a few different actions taken involving variables:
1. **declaring** - when you set or declare the *data type* and *name* of the variable. These two pieces of a variable do *not* change.
1. **assigning** - when you set the *value* of the variable. The value of a variable *can* change.
1. **accessing** - when you retrieve the *value* of the variable by calling it's *name*.

You *must* declare and assign a variable before you can access it.

Take a look at the visualizer on the left to see an example of how this works.

Data Types - Integers

Integers

Integers (often called `ints`) are whole numbers. They can be positive or negative. Do not use a comma when typing large numbers

Enter the code below and click the `TRY IT` button.

```
int number = 50;
System.out.print(number);
```

▼ **5 vs. “5”**
`5` is not the same thing as `“5”`. The first one is an integer, the second is a string. You will see in a later lesson the different operations you can perform on strings and numbers. Treating a string as a number can cause errors.

challenge

What happens if you:

- Change the variable to `5000`?
- Change the variable to `5,000`?
- Change the variable to `050`?

Data Types - Floating Point Numbers

Floating Point numbers

Floating point numbers (often called floats) are numbers with a decimal. They can be positive or negative.

```
double fraction = 0.5;
System.out.print(fraction);
```

▼ **Why is float called double?**
In Java, there is a data type called **float** but as it only uses 4 bytes it is insufficient for most math. Instead, we use **double** which uses 8 bytes or double the space of a float

challenge

What happens if you:

- Change the variable to `50.`?
- Change the variable to `.001`?

Data Types - Boolean

Boolean

Boolean values mean `true` or `false`. You will see how boolean values are used when we talk about conditionals and while loops.

```
boolean thisIsFun = true;
System.out.print(thisIsFun);
```

challenge

What happens if you:

- Change the variable to `false`?
- Change the variable to `True`?
- Change the variable to `False`?

Data Types - Strings

Strings

A string is a collection of text, numbers, or symbols. Strings are always surrounded by quotation marks.

```
String words = "This is a string";
System.out.print(words);
```

challenge

What happens if you:

- Use single (') quotation marks?
- Forget one of the quotation marks?
- Forget both quotation marks?

Notice that when you print a string, the quotation marks are not printed.

Declaring Variables

Declaring a Variable

Declaring a variable has two parts - setting or declaring the *data type* and the *name* of the variable. These two pieces of a variable do *not* change.

To declare a variable, type the data type and name of the variable you want to create, and `;` (semi-colon).

```
String my_variable;
```

You will notice we are not printing anything - that is because no value has been assigned yet. The declaration step sets aside empty memory.

challenge

What happens if you:

- Create two variables with the same name?
- Create two varaibles with the same name but different capitalization (ie `var` and `Var`)?
- Create two variables of different types with the same name?

Variable Naming Rules

Here are the rules for naming a variable.

Rule	Correct	Incorrect
Start with a letter, dollar sign or underscore	`variable`, `$variable`, `_variable`	`1variable`
Remainder of variable name is letters, numbers, or	`var_i_able`, `var1able`	`var-i-able`, `var!able`

underscores		
Cannot use a Java keyword	`my_class`	`class`
Variables are case sensitive	`variable`, `Variable`, and `VARIABLE` are all different variables	

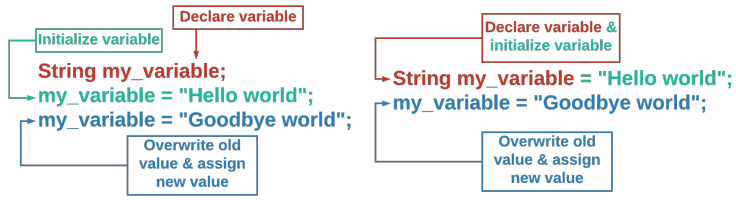
▼ **What are the Java key words?**

abstract	continue	for	new	switch
assert	default	goto	package	synchronized
boolean	do	if	private	this
break	double	implements	protected	throw
byte	else	import	public	throws
case	enum	instanceof	return	transient
catch	extends	int	short	try
char	final	interface	static	void
class	finally	long	strictfp	volatile
const	float	native	super	while

Initializing, Assigning, and Accessing

Initializing Assigning Values

We call the process of setting the **initial** value of a variable **initialization**. You can do this separately after the declaration or combine it into the same statement as the declaration.



.guides/img/VariableAssignment2

Since the value stored in a variable can change, we call changing the value **assigning** or **re-assigning**. Use the assignment operator to give a variable a new value.

Accessing Variables

Enter the code below and see the results of the `println` commands. Use the code visualizer to see how the value of `my_variable` changes.

```
String my_variable = "Hello world";
System.out.println(my_variable);
my_variable = "Goodbye world";
System.out.println(my_variable);
```

When we use a variable's name to get the value like in the `println` statements above, we say we are **accessing** the variable.

Code Visualizer