# Variables

A Variable is basically a placeholder that can be used in your code.

Note the naming convention that we use for variable names:

* playerName
* player1StartPoint
* enemyLives
* countDownTimer
* exitPointOfTheLevel

The standard we use for naming variables is called **lower camelcase**.

* The name starts with a lowercase letter
* All sequential words have upper case letters as their starting letter
* There are no spaces or underscores
* Variables can use numbers but not as their first character

## Common types of variables:

### Integer: (int)

**Whole numbers**

4

2211

834

-4

### Float: (float)

**Decimal numbers**

3.0

24.5

-3.6

2.52453545

### String: (string)

**Words and numbers that are not used for calculations**

“Hello”

“Green “

“4”

“223.df33”

### Boolean: (bool)

**True or False**

True

False

### Vector3: (Vector3)

**A series of three floats that are used mainly for position, rotation, force,direction or scale**

0,4,6

3.67,-34,30

0,0,0

-77,-78,-79

## Declaring Variables

Variables are normally declared at the beginning of a script.

When declaring variables you can delaire them as **public** or **private**

public class HelloWorld : MonoBehaviour {

public float playerHealth;

public int playerLives = 5;

public bool isPlayerAlive;

private Vector3 playerStartPosition;

private string playerName;

void Start()

{

}

}

If the above script is attached to an object you will see this in the inspector when that object is selected.



Notice that only the variables that have been declared as public are visible, and that playerLives has defaulted to 5 while everything else falls back to a default value of 0 or false;

# References

You can also store references to other objects or components in the game.

The standard reference is a GameObject reference as everything in a unity game is a GameObject.

Here is an example script that will reference the player game object.

public class HelloWorld : MonoBehaviour {

public GameObject playerReference;

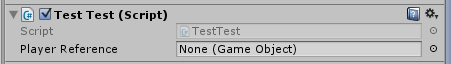
void Start (){

playerReference.transform.position = new Vector3 (0, 0, 0);

}

}

If you attach this script to an object in the scene you will see that there is a slot in the Inspector that supports a Game Object now.



Once you have dragged a game object in the scene to this slot, the game, this script will run and move position of that referenced game object will move to position 0,0,0.

## Why is this useful?

This is a quickest way of letting a script know about objects in the game other than the object that the script is attached to.

You could reference the player on an enemy script so that the enemy always faces and moves towards the player.

You could reference a score counter on the UI from a player script, so that when a player picks up a coin the script can tell the score counter to increase

## Further information

<https://unity3d.com/learn/tutorials/modules/beginner/scripting/variables-and-functions?playlist=17117>