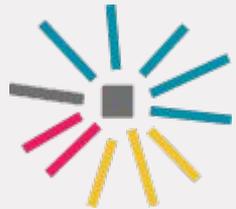


CODE BASICS

PALIMPSEST



CODE BASICS

What *is* all that stuff?

- Definition
- Assignment
- Comments
- Braces
- Code Block
- EXERCISE

CODE BASICS



There is a tradition in programming...



“HELLO WORLD”



Hello.

“HELLO WORLD”



Hello.

THAT'S YOU!

“HELLO WORLD”

**It looks
something
like this...**

HELLO WORLD

```
// Class to express myself to the world.  
  
public class YoWorldIExist  
{  
    // Method to print string to console.  
    public void Speak( string phrase )  
    {  
        Debug.Log( phrase );  
    }  
}
```

HELLO WORLD

```
// Class to express myself to the world.  
  
public class YoWorldIExist {  
    // Method to print string to console.  
    public void speak( String phrase )  
    {  
        System.out.println( phrase );  
    }  
}
```

**There's a LOT of
vocabulary here.**

HELLO WORLD

```
// Class to express myself to the world.  
  
public class YoWorldIExist  
{  
    // Method to print string to console.  
    public void Speak( string phrase )  
    {  
        Debug.Log( phrase );  
    }  
}
```

HELLO WORLD

```
// Class to express myself to the world.  
  
public class YoWorldIExist  
{  
    // Method to print string to console.  
    public void Speak( string phrase )  
    {  
        Debug.Log( phrase );  
    }  
}
```

HELLO WORLD

```
// Class to express myself to the world.  
  
public class YoWorldIExist  
{  
    // Method to print string to console.  
    public void Speak( string phrase )  
    {  
        Debug.Log( phrase );  
    }  
}
```

HELLO WORLD

```
// Class to express myself to the world.  
  
public class YoWorldIExist  
{  
    // Method to print string to console.  
    public void Speak( string phrase )  
    {  
        Debug.Log( phrase );  
    }  
}
```

HELLO WORLD

```
// Class to express myself to the world.  
  
public class YoWorldIExist  
{  
    // Method to print string to console.  
    public void Speak( string phrase )  
    {  
        Debug.Log( phrase );  
    }  
}
```

HELLO WORLD

```
// Class to express myself to the world.  
  
public class YoWorldIExist  
{  
    // Method to print string to console.  
    public void Speak( string phrase )  
    {  
        Debug.Log( phrase );  
    }  
}
```

HELLO WORLD

```
// Class to express myself to the world.  
public class WorldIs {  
    {  
        // Method to print string to console.  
        public void say ( string phrase )  
        {  
            Debug.Log ( phrase );  
        }  
    }  
}
```

**Even the
punctuation is
important.**

HELLO WORLD

```
// Class to express myself to the world.  
  
public class YoWorldIExist  
{  
    // Method to print string to console.  
    public void Speak( string phrase )  
    {  
        Debug.Log( phrase );  
    }  
}
```

HELLO WORLD

**We'll talk about
all of it.**

```
// Class to express myself to the world.
```

```
public class YoWorldExist
```

```
// Method to print string to console.
```

```
public void Speak( string phrase )
```

```
{
```

```
    Debug.Log( phrase );
```

```
}
```

```
}
```

HELLO WORLD

```
// Class to express myself to the world.  
  
public class YoWorldIExist  
{  
    // Method to print string to console.  
    public void Speak( string phrase )  
    {  
        Debug.Log( phrase );  
    }  
}
```

HELLO WORLD

```
// Class to express myself to the world.  
  
public class YoWorldIExist  
{  
    // Method to print string to console.  
    public void Speak( string phrase )  
    {  
        Debug.Log( phrase );  
    }  
}
```

Not an oversight.

COMMENTS

```
// Words after two forward slashes are  
// ignored by the compiler.
```

```
/* To write long or multi-line comments  
use a forward slash and asterisk paired.
```

```
End the multi-line comment with it  
reversed. Like this... */
```

SAY THE UNCOMMENTED WORDS

```
// Lying, thinking  
// Last night  
How to find my soul a home  
Where water is not thirsty  
And bread loaf is not stone  
/* I came up with one thing  
And I don't believe I'm wrong  
That nobody,  
But nobody  
Can make it out here alone. */
```

Maya Angelou

COMMENTS

```
// Are notes for programmers
```

```
/* Use them to explain something to  
someone coming after you.
```

```
Or to make notes for Future You. */
```

How does the computer read?

Just like us.



Top to bottom.



Left to right.



How does the
computer read?

EXCEPT

It will jump to other parts of code when directed.

We'll get to that later.

CLASS DECLARATION

```
// Class to express myself to the world.

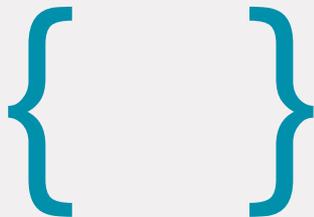
public class YoWorldIExist
{
    // Method to print string to console.
    public void Speak( string phrase )
    {
        Debug.Log( phrase );
    }
}
```

CLASS DECLARATION

```
// Class to express myself to the world.  
  
public class YoWorldIExist  
{  
    // Method to print string to console.  
    public void Speak( string phrase )  
    {  
        Debug.Log( phrase );  
    }  
}
```

CURLY BRACES

- Always paired



```
// Class to express myself to the world.  
  
public class YoWorldIExist  
{  
    // Method to print string to console.  
    public void Speak( string phrase )  
    {  
        Debug.Log( phrase );  
    }  
}
```

CURLY BRACES

- Always paired
- 'Contain' a block of code.
- What do **these** braces contain?

```
// Class to express myself to the world.  
  
public class YoWorldIExist  
{  
    // Method to print string to console.  
    public void Speak( string phrase )  
    {  
        Debug.Log( phrase );  
    }  
}
```

CURLY BRACES

- Always paired
- 'Contain' a block of code.
- What about **these**?

```
// Class to express myself to the world.

public class YoWorldIExist
{
    // Method to print string to console.
    public void Speak( string phrase )
    {
        Debug.Log( phrase );
    }
}
```

OTHER 'CONTAINERS'

- There are two more 'containers'
- Any guesses?

```
// Class to express myself to the world.  
  
public class YoWorldIExist  
{  
    // Method to print string to console.  
    public void Speak( string phrase )  
    {  
        Debug.Log( phrase );  
    }  
}
```

PARENTHESES

- There are two more 'containers'
- Any guesses?

```
// Class to express myself to the world.  
  
public class YoWorldIExist  
{  
    // Method to print string to console.  
    public void Speak( string phrase )  
    {  
        Debug.Log( phrase );  
    }  
}
```

THE TERMINATOR

```
public class IHaveLoudCats
{
    // The Start method runs once at the start of the program.
    public void Start()
    {
        // This next statement assignment statement.
        string firstPhrase = "My cats are super loud rn.";

        // The next statement uses the Debug method's Log
        // function to display firstPhrase.
        Debug.Log ( firstPhrase );
    }
}
```

THE TERMINATOR



```
public class IHaveLoudCats
{
    // The Start method runs once at the start of the program.
    public void Start()
    {
        // This next statement assignment statement.
        string firstPhrase = "My cats are super loud rn.";

        // The next statement uses the Debug method's Log
        // function to display firstPhrase.
        Debug.Log ( firstPhrase );
    }
}
```

THE TERMINATOR

- Which punctuation seems like a terminator?

```
public class IHaveLoudCats
{
    // The Start method runs once at the start of the program.
    public void Start()
    {
        // This next statement assignment statement.
        string firstPhrase = "My cats are super loud rn.";

        // The next statement uses the Debug method's Log
        // function to display firstPhrase.
        Debug.Log ( firstPhrase );
    }
}
```

THE TERMINATOR

- Which punctuation seems like a terminator?

```
public class IHaveLoudCats
{
    // The Start method runs once at the start of the program.
    public void Start()
    {
        // This next statement assignment statement.
        string firstPhrase = "My cats are super loud rn.";

        // The next statement uses the Debug method's Log
        // function to display firstPhrase.
        Debug.Log ( firstPhrase );
    }
}
```

THE TERMINATOR

- How many terminators here?

```
public class IHaveLoudCats
{
    // The Start method runs once at the start of the program.
    public void Start()
    {
        // This next statement assignment statement.
        string firstPhrase = "My cats are super loud rn.";

        // The next statement uses the Debug method's Log
        // function to display firstPhrase.
        Debug.Log ( firstPhrase );
    }
}
```

THE TERMINATOR

- Why semi-colon?
- Because period already has a lot of other jobs.

```
public class IHaveLoudCats
{
    // The Start method runs once at the start of the program.
    public void Start()
    {
        // This next statement assignment statement.
        string firstPhrase = "My cats are super loud rn.";

        // The next statement uses the Debug method's Log
        // function to display firstPhrase.
        Debug.Log ( firstPhrase );
    }
}
```

CODE BLOCK

Any code enclosed between braces is called a block.

How many blocks are here?

```
public void Start()  
{  
    string firstPhrase = "My cats are super  
loud rn.";  
    Speak( firstPhrase );  
}  
  
public void Speak( string phrase )  
{  
    Debug.Log( phrase );  
}
```

CODE BLOCK

Any code enclosed between braces is called a block.

How many blocks are here?

```
public void Start()  
{  
    string firstPhrase = "My cats are super  
loud rn.";  
    Speak( firstPhrase );  
}  
  
public void Speak( string phrase )  
{  
    Debug.Log( phrase );  
}
```

VARIABLES



- A small piece or “chunk” of data
- Temporary
- Must be **declared** then **assigned**

VARIABLES



HAVE DATA TYPES

DATA TYPES



INTEGER: positive and negative whole number (1, -77, 25)

FLOATING POINT: decimal number (3.14f, 10.25f)

CHAR: a single *character* ('c', 'e', 'l')

BOOLEAN: a state holder, used for logic (true, false, 1, 0)

AND MORE!

DATA TYPES



Like shapes in **Perfection**, data types have to match.

VARIABLES HAVE



- A name, a type, and a value
- A specific location in memory
- A variable's value can just be referenced or replaced
(names and types don't change)



VARIABLE DECLARATION

Hey, Memory. I need a variable.

Type? It's a phone number. So, integer. Int. Yeah.

Call it, um... how about `phoneNumber`. I know, right?

I'm not sure what number yet, can you just hang on to it?

Thanks.

Before you use a variable, you must **declare** it.



VARIABLE DECLARATION

What are the data types?

```
// Before you use a variable, you must  
// declare it.
```

What are the names?

```
int songNumber;
```

Where are the terminators?

```
float songLength;
```

```
string songTitle;
```

VARIABLE ASSIGNMENT

What is the value of
songNumber?

```
// Before you use a variable, you must  
// declare it.
```

Why does songLength's
value have an f?

```
songNumber = 4;  
  
songLength = 180.4f;
```

What is different about
songTitle's value?

```
songTitle = "Magnificent (She Says)";
```

VARIABLE ASSIGNMENT

The equal sign has a special job here.

It **assigns** the value on the **right** to the value on the **left**.

It *does not* show equivalence.

What is the value of z?

```
// Read this as 'x is assigned the value 5'  
// Or 'y gets the value 10'
```

```
int x = 5;
```

```
int y = 10;
```

```
int z;
```

```
// You can use variables as the values of  
// other variables. This is REALLY useful.
```

```
z = x + y;
```



EXERCISE

Open up a
script in
Unity



EXERCISE

Answer

1. What is the class name?
2. What variables are declared? What are their names and types?
3. How many methods are there? What are their names?
4. How many statements are there? How many of the statements are assignments?
5. Delete a semi-colon. What happened? What type of error is this?

NEXT UP

Hardware

