Values :

- Values are immutable
- create new values, instead of changing them

- Object class

```
object Values {
      def main(args: Array[String]): Unit = {
      val favoriteDessert = "Pumpkin Pie";
      println(favoriteDessert);

OUTPUT:
Pumpkin Pie
```

Reminder: Her Videos may need tweaking to the code just an FYI !!!

Use GeeksforGeeks or chatgpt to figure out issues!!!!

After println or Any print no need for semicolon ;(

Types:

- **Everything In Scala Has a type**
- **has Static type system**
- **catches errors at compile time**

How to Define a Function

```
      object Functions{
            def main(args: Array[String]): Unit =   {
      println(sum(2,2));
      println(lightSwitch(true);
      println(lightSwitch(false);
      def sum(n1: Int, n2, Int): Int = n1+n2

      def lightSwitch(switchon: Boolean): String = if (switchOn) "light on!" else "light
off!;

OUTPUT:
4
light on !
light off !
```

Challenge 2

```
object Challenge2 {
  def main(Args: Array[String]): Unit = {
    // Write a function called product which takes two parameters of type int and multiply them
together
    def product(parum: Int, parum2: Int, parum3: Int): Int = parum * parum2 * parum3;
    println(product(3,29, 19))

    /*
    Write a function called greeting, which takes a parameter called Name of type String and
outputs
    a String; the parameter should have a default value, so that an appropriate String is
outputted if the
    function is called without a parameter.
     */
    println(greetingPerson(greets = false))
    println(greetingPerson(greets = true))
    def greeting(n1: String): String = "David";
    def greetingPerson(greets: Boolean = false): String = if (greets) "Hello David, How is your
day?"
                                          else "You no welcome.... >;["
  }
}
```

I have did the second challenge a bit different since I was trying to figure out on my own; the programmer does it a bit different or follows a bit better the instructions I just wanted to reinforce the previous material and continue to explore other options