EXAMPLE 29: STRINGS AND NUMBERS -BUT NOT OBJECTS - ARE 'PASSED-BY-VALUE' TO FUNCTIONS

EXAMPLE 29: STRINGS AND NUMBERS - BUT NOT OBJECTS - ARE 'PASSED-BY-VALUE' TO FUNCTIONS

IN OTHER WORDS, A COPY IS MADE OF STRING OR NUMBER VARIABLES

AND THE FUNCTION WORKS WITH THE COPY - NOT THE ORIGINAL.

EXAMPLE 29: STRINGS AND NUMBERS - BUT NOT OBJECTS - ARE 'PASSED-BY-VALUE' TO FUNCTIONS

IN OTHER WORDS, A COPY IS MADE OF STRING OR NUMBER AND THE FUNCTION WORKS WITH THE COPY - NOT THE ORIGINAL.

EVEN IF THE FUNCTION MODIFIES THE VARIABLE, THE ORIGINAL IS UNCHANGED.

EVEN IF THE FUNCTION MODIFIES THE VARIABLE, THE ORIGINAL IS UNCHANGED.

```
function passByValueExample() {
  var x = 10;
  var y = "Vitthal";
  console.log("Initial values (in calling function) = " + x + " " + y);
  modifyX(x,y);
  console.log("Final values (in calling function) = " + x + " " + y);
}

function modifyX(someNumber,someString) {
  console.log("Values passed into function = " + someNumber + " and " +
someString);
  someNumber = someNumber + 10;
  someString = "HumptyDumpty";
  console.log("Values passed into function have been modified to = " +
someNumber + " and " + someString);
}
```

EVEN IF THE FUNCTION MODIFIES THE VARIABLE, THE ORIGINAL IS UNCHANGED.

```
function passByValueExample() {
  var x = 10;
  var y = "Vitthal";
  console.log("Initial values (in calling function) = " + x + " " + y);
  modifyX(x,y);
  console.log("Final values (in calling function) = " + x + " " + y);
}

function modifyX(someNumber,someString) {
  console.log("Values passed into function = " + someNumber + " and " +
someString);
  someNumber = someNumber + 10;
  someString = "HumptyDumpty";
  console.log("Values passed into function have been modified to = " +
someNumber + " and " + someString);
}
```

EVEN IF THE FUNCTION MODIFIES THE VARIABLE, THE ORIGINAL IS UNCHANGED.

```
function passByValueExample() {
  var x = 10;
  var y = "Vitthal";
  console.log("Initial values (in calling function) = " + x + " " + y);
  modifyX(x,y);
  console.log("Final values (in calling function) = " + x + " " + y);
}

function modifyX(someNumber,someString) {
  console.log("Values passed into function = " + someNumber + " and " +
someString);
  someNumber = someNumber + 10;
  someString = "HumptyDumpty";
  console.log("Values passed into function have been modified to = " +
someNumber + " and " + someString);
}
```

TWO VARIABLES, ONE A NUMBER, THE OTHER IS A STRING.

EVEN IF THE FUNCTION MODIFIES THE VARIABLE, THE ORIGINAL IS UNCHANGED.

```
function passByValueExample() {
  var x = 10;
  var y = "Vitthal";
  console.log("Initial values (in calling function) = " + x + " " + y);
  modifyX(x,y);
  console.log("Final values (in calling function) = " + x + " " + y);
}

function modifyX(someNumber,someString) {
  console.log("Values passed into function = " + someNumber + " and " +
someString);
  someNumber = someNumber + 10;
  someString = "HumptyDumpty";
  console.log("Values passed into function have been modified to = " +
someNumber + " and " + someString);
}
```

THE VARIABLES ARE PASSED INTO A FUNCTION..

EVEN IF THE FUNCTION MODIFIES THE VARIABLE, THE ORIGINAL IS UNCHANGED.

AND THAT FUNCTION MODIFIES THE VALUES OF THE VARIABLES PASSED IN

```
function passByValueExample() {
  var x = 10;
  var y = "Vitthal";
  console.log("Initial values (in calling function) = " + x + " " + y);
  modifyX(x,y);
  console.log("Final values (in calling function) = " + x + " " + y);
}

function modifyX(someNumber,someString) {
  console.log("Values passed into function = " + someNumber + " and " +
someString);
    someNumber = someNumber + 10;
    someString = "HumptyDumpty";
  console.log("Values passed into function have been modified to = " +
someNumber + " and " + someString);
}
```

EVEN IF THE FUNCTION MODIFIES THE VARIABLE, THE ORIGINAL IS UNCHANGED.

```
function passByValueExample() {
  var x = 10;
  var y = "Vitthal";
  console.log("Initial values (in calling function) = " + x + " " + y);
  modifyX(x,y);
  console.log("Final values (in calling function) = " + x + " " + y);
}

function modifyX(someNumber,someString) {
  console.log("Values passed into function = " + someNumber + " and " +
someString);
  someNumber = someNumber + 10;
  someString = "HumptyDumpty";
  console.log("Values passed into function have been modified to = " +
someNumber + " and " + someString);
}
```

EVEN IF THE FUNCTION MODIFIES THE VARIABLE, THE ORIGINAL IS UNCHANGED.

```
Initial values (in calling function) = 10 Vitthal
Values passed into function = 10 and Vitthal
Values passed into function have been modified to = 20 and HumptyDumpty
Final values (in calling function) = 10 Vitthal
```

EVEN IF THE FUNCTION MODIFIES THE VARIABLE, THE ORIGINAL IS UNCHANGED.

```
Initial values (in calling function) = 10 Vitthal
Values passed into function = 10 and Vitthal
Values passed into function have been modified to = 20 and HumptyDumpty
Final values (in calling function) = 10 Vitthal
```

EVEN IF THE FUNCTION MODIFIES THE VARIABLE, THE ORIGINAL IS UNCHANGED.

```
Initial values (in calling function) = 10 Vitthal
Values passed into function = 10 and Vitthal
Values passed into function have been modified to = 20 and HumptyDumpty
Final values (in calling function) = 10 Vitthal
```

EVEN IF THE FUNCTION MODIFIES THE VARIABLE, THE ORIGINAL IS UNCHANGED.

```
Initial values (in calling function) = 10 Vitthal
Values passed into function = 10 and Vitthal
Values passed into function have been modified to = 20 and HumptyDumpty
Final values (in calling function) = 10 Vitthal
```