

Solution Review: Defining Variables

This lesson gives a detailed solution review of the challenge in the previous lesson.

We'll cover the following ^

- Solution:
- Explanation

Solution:

```
fn test() {  
  // declare a mutable variable `x`  
  let mut x = 1000;  
  // declare a variable `y`  
  let y="Programming";  
  // print output of `x`  
  println!("x:{}", x);  
  // print output of `y`  
  println!("y:{}", y);  
  // update x  
  x = 1100;  
  // print output of `x`  
  println!("x:{}", x);  
  // print output of `y`  
  println!("y:{}", y);  
}
```



Explanation

- On **line 3**, a **mutable variable** `x` is defined and assigned the value `1000`.
- On **line 5**, a **variable** `y` is defined and assigned the value `Programming`.
- On **line 7 and line 9**, the value of `x` and `y` is printed respectively.
- On **line 11**, the value of `x` is updated since `x` is mutable.
- On **line 13 and line 15**, the values of `x` and `y` is printed respectively.

```
fn test() {  
    let mut x = 1000;  
    let y = "Programming";  
    println!("x:{}", x);  
    println!("y:{}", y);  
    x = 1100;  
    println!("x:{}", x);  
    println!("y:{}", y);  
}
```

Output

1 of 9

```
fn test() {  
    let mut x = 1000;  
    let y = "Programming";  
    println!("x:{}", x);  
    println!("y:{}", y);  
    x = 1100;  
    println!("x:{}", x);  
    println!("y:{}", y);  
}
```

Output

2 of 9

```
fn test() {  
    let mut x = 1000;  
    let y = "Programming";  
    println!("x:{}", x);  
    println!("y:{}", y);  
    x = 1100;  
    println!("x:{}", x);  
    println!("y:{}", y);  
}
```

Output

3 of 9

```
fn test() {  
    let mut x = 1000;  
    let y = "Programming";  
    println!("x:{}", x);  
    println!("y:{}", y);  
    x = 1100;  
    println!("x:{}", x);  
    println!("y:{}", y);  
}
```

Output

x: 1000

4 of 9

```
fn test() {  
    let mut x = 1000;  
    let y = "Programming";  
    println!("x:{}", x);  
    println!("y:{}", y);  
    x = 1100;  
    println!("x:{}", x);  
    println!("y:{}", y);  
}
```

Output

```
x: 1000  
y: Programming
```

5 of 9

```
fn test() {  
    let mut x = 1000;  
    let y = "Programming";  
    println!("x:{}", x);  
    println!("y:{}", y);  
    x = 1100; x is mutable variable so value can be changed  
    println!("x:{}", x);  
    println!("y:{}", y);  
}
```

Output

```
x: 1000  
y: Programming
```

6 of 9

```
fn test() {  
    let mut x = 1000;  
    let y = "Programming";  
    println!("x:{}", x);  
    println!("y:{}", y);  
    x = 1100;  
    println!("x:{}", x);  
    println!("y:{}", y);  
}
```

Output

```
x: 1000  
y: Programming  
x: 1100
```

7 of 9

```
fn test() {  
    let mut x = 1000;  
    let y = "Programming";  
    println!("x:{}", x);  
    println!("y:{}", y);  
    x = 1100;  
    println!("x:{}", x);  
    println!("y:{}", y);  
}
```

Output

```
x: 1000  
y: Programming  
x: 1100  
y: Programming
```

8 of 9

```
fn test() {  
    let mut x = 1000;  
    let y = "Programming";  
    println!("x:{}", x);  
    println!("y:{}", y);  
    x = 1100;  
    println!("x:{}", x);  
    println!("y:{}", y);  
}end of program
```

Output

```
x: 1000  
y: Programming  
x: 1100  
y: Programming
```

9 of 9

—

[]

Now you have learned the concept of variables, what they are, and how to create them. What if you want to define the type of value that goes inside the variable? Let's learn about this in the next chapter "Data Types".