

Forced Unwrapping

To unwrap or retrieve the value of an optional, place an exclamation mark (!) after

Example: Write a program to calculate a person's weekly salary. The person is paid \$750 per week for 40 hours of work. If the person works more than 40 hours per week, their overtime hours are billed at \$23.50 / hour.

```
var overtimeHours: Int?  
// ... some code goes here  
let salary:Double = 750 + overtimeHours! * 23.50
```

Forced unwrapping is very dangerous because it relies on the developer being **absolutely certain** that the variable will contain a value **at the time the variable is accessed**.

Instead of using force unwrapping, you can use:

- Optional Binding: if-let statements
 - Used only in the *body* of a program
- Guard Let: Early exit
 - Used only in a function
- Nil coalescing: Provides a default value to your optional

Optional Binding

Example of the if-let statement

```
var overtimeHours: Double? = 30;
if let ot = overtimeHours {
    let salary:Double = 750 + ot * 23.50;
    print(salary)
}
else {
    let salary:Double = 750
    print(salary)
}
```

This is the same as saying:

```
var overtimeHours: Double? = 30;
if overtimeHours == nil {
    let ot = overtimeHours!
    let salary:Double = 705 * ot * 23.50
}
else {
    let salary:Double = 750
    print(salary)
}
```

It is also possible to perform optional binding on several variables at the same time:

```
if let tmp1 = optional1 {  
    if let tmp2 = optional2 {  
        if let tmp3 = optional3 {  
        }  
    }  
}  
  
// shortcut  
  
if let tmp1 = optional1, let tmp2 = optional2, let tmp3 = optional3 {  
}
```

Nil coalescing:

Swift's nil coalescing operator will either:

1. unwrapping an optional if it has a value, or,
2. providing a default if the optional is empty.

```
let middleName:String? = ...  
let unwrappedMiddleName:String = middleName ?? "Missing";  
print(unwrappedMiddleName);
```

The nil coalescing operator is shorthand syntax for doing a null check with a ternary operator:

```
var nameC = optionalA != nil ? optionalA! : defaultName
```