

Progress Bar Introduction

Progress bar, formally known as `UIProgressView` in swift 4.1, is a common element of a page. It is used to indicate the progress of a **computable** process, such as downloading a file. It should not be used for **incomputable** process, such as attempting to establish Internet connection or searching for a content (for such use, please see `spinner` to the right.)

Image of Progress Bar



Useful Class Method

```
func setProgress(_ progress: Float, animated: Bool) {
    // A function to set progress of the progress bar.
    // The first positional argument is within range 0.0 to 1.0 inclusive, indicating the progress of the task.
    // The second argument animated indicates whether the change in progress should be animated. When set to false, the progress bar will "jump" to the updated progress. Otherwise, it will be filled/unfilled gradually to the updated progress.
    progressBar.setProgress(progress, animated: animated)
}
```

Useful Instance Variables

`isHidden`
A `Bool` determines whether the progress bar is hidden or not.

`progress`
A `Float` within range 0.0 to 1.0 inclusive, indicating the progress.

MCVE (Minimal Complete Verifiable Example)

```
/*\ to run this example, simply create up a new project, then
| | copy and paste the code below to ViewController.swift
| | create a button and a progress bar in the storyboard, then
```

MCVE (Minimal Complete Verifiable Example) (cont)

```
> | | create a reference outlet from the progress bar to
| | progressBar variable, then create
| | an action outlet from the button to startTask() function
\*/
import UIKit
class ViewController: UIViewController {
    @IBOutlet weak var progressBar: UIProgressView!
    //outlet of the progressBar

    override func viewDidLoad() {
        super.viewDidLoad()
        //set the progressBar to 0, default value is 0.5
        progressBar.setProgress(0, animated: false)
        //hide the progressBar
        progressBar.isHidden=true
        var t:Timer!
        @objc func updateProgress(){
            // an example function to increase the progress of the task by 1%
            // only called by timer in the following function
            // @objc decorator exposes the function to timer
            progressBar.setProgress(progressBar.progress+0.01,animated: true)
            if progressBar.progress>=1.0{
                t.invalidate() // stop the timer
                progressBar.isHidden=true // hide the progress bar
                progressBar.setProgress(0, animated: false) // reset the progress bar
            }
        }

        @IBAction func startTask(_ sender: Any) {
            // outlet of the button that activates this function
            // an example function that increases the progress of
```

MCVE (Minimal Complete Verifiable Example) (cont)

```
> // progress bar by 0.1 every 0.05 second (simulating a
// task that takes 5 seconds to complete)
progressBar.isHidden=false // display the progress bar
t=Timer.scheduledTimer(timeInterval: 0.05, target: self, selector: #selector(updateProgress), userInfo: nil, repeats: true) // set the timer
}
```

Spinner Introduction

Spinner, formally known as `UIActivityIndicatorView` in swift 4.1, is a common element for an application. It is used to indicate that there is an ongoing process (either executing in the front or background) that consumes unknown amount of time, such as searching for a specific content or making computation.

Image of Spinner

