



TECH 4 FREELANCERS

TECHNOLOGY TO EMPOWER YOU IN YOUR WORK

[HOME](#)[HARDWARE](#) ▾[SOFTWARE](#)[GUIDES & TIPS](#) ▾[STATIONERY](#)[OPINIONS](#)[ABOUT](#)[CONTACT](#)

echo off vs @echo off: What's the difference?

🕒 27th July 2017 👤 Andrea Luciano Damico 📁 Guides & Tips, Windows 💬 0

```
jQuery(document).ready(function() {  
    $('.panel-btn').on('click', function(event) {  
        event.preventDefault();  
        $('.panel').addClass('is-visible');  
    });  
    $('.panel').on('click', function(event) {  
        if (event.target.is('.panel')) {  
            $('.panel').removeClass('is-visible');  
            event.preventDefault();  
        }  
    });  
});  
  
jQuery(document).ready(function() {  
    //open the lateral panel  
    $('.panel-btn').on('click', function(event) {  
        event.preventDefault();  
        $('.panel').addClass('is-visible');  
    });  
    //close the lateral panel  
    $('.panel').on('click', function(event) {  
        if (event.target.is('.panel')) {  
            $('.panel').removeClass('is-visible');  
            event.preventDefault();  
        }  
    });  
});
```

People with at least a passing interest in batch scripting may have noticed that all scripts begin with the same line:

@echo off

The more technically inclined reader already knows that this command disables outputting command names in the Command Prompt window, but may still wonder why there is a leading @

RECENT ARTICLES

Amazon Basics stick ballpoint pen review

7th September 2017

Is mini ITX worth it?

31st August 2017

What I would like to see in the Raspberry Pi 4 and why it won't happen

28th August 2017

How to pick colours in MacOS

24th August 2017

Configuring a swap file in Linux

21st August 2017

LIKE US ON FACEBOOK

sign. The best way to find out is to see a script in action. We won't be showing something that pushes the boundaries of reality: it's a simple Hello World program. Our aim here is to show how that script will look inside a Command Prompt window in the cases where we don't use `echo off` at all, we use `echo off` and we use `@echo off`.

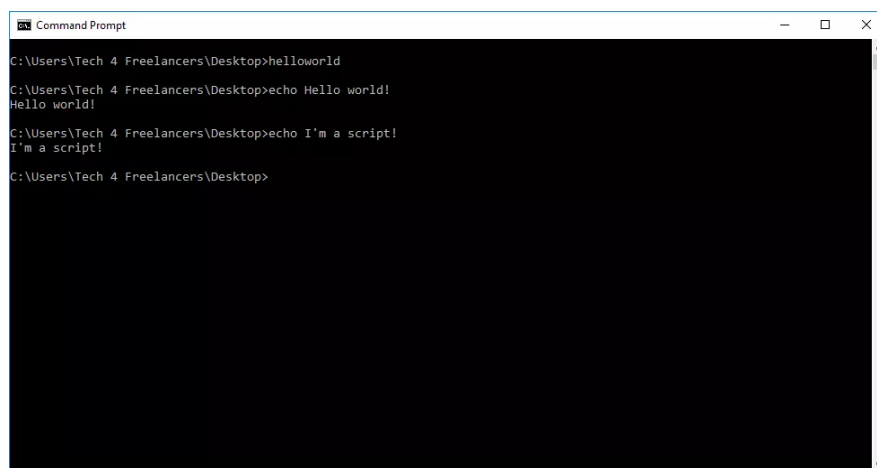
Case #1: No `echo off` at all

Our sample script only contains two lines:

```
echo Hello World!
```

```
echo I'm a script!
```

When we run it, this is what happens:



```
Command Prompt
C:\Users\Tech 4 Freelancers\Desktop>helloworld
C:\Users\Tech 4 Freelancers\Desktop>echo Hello world!
Hello world!
C:\Users\Tech 4 Freelancers\Desktop>echo I'm a script!
I'm a script!
C:\Users\Tech 4 Freelancers\Desktop>
```

Whenever the program is run, not only does the Command Prompt show every command that's being run, but it still displays the active folder. As you can imagine, this isn't very readable, for relevant information may get lost in the tens of lines of output.

Case #2: Using `echo off`

Let's edit our sample script by adding `echo off` at the beginning. Our code is as follows:

```
echo off
```

```
echo Hello World!
```

```
echo I'm a script!
```

Let's see what happens when we run it:



Tech 4 Freelancers
169 likes

Like Page

Be the first of your friends to like this



YOUR T4F EXPERIENCE

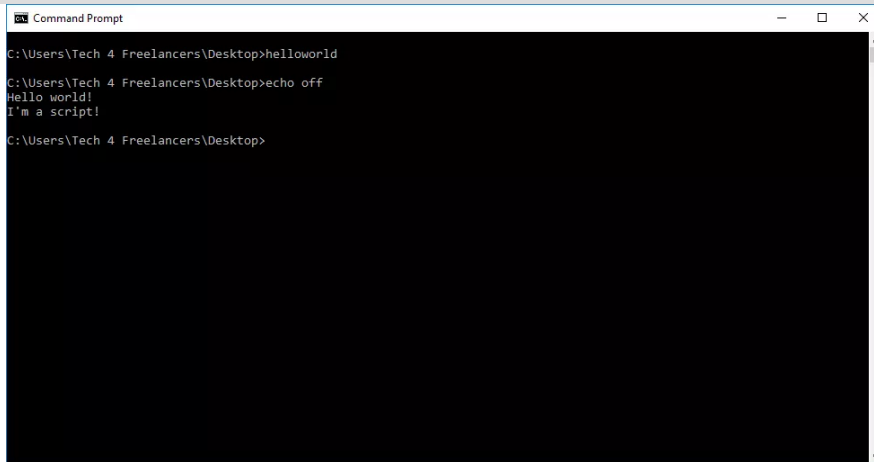
Register

Log in

Entries [RSS](#)

Comments [RSS](#)

WordPress.org



```
C:\Users\Tech 4 Freelancers\Desktop>helloworld
C:\Users\Tech 4 Freelancers\Desktop>echo off
Hello world!
I'm a script!
C:\Users\Tech 4 Freelancers\Desktop>
```

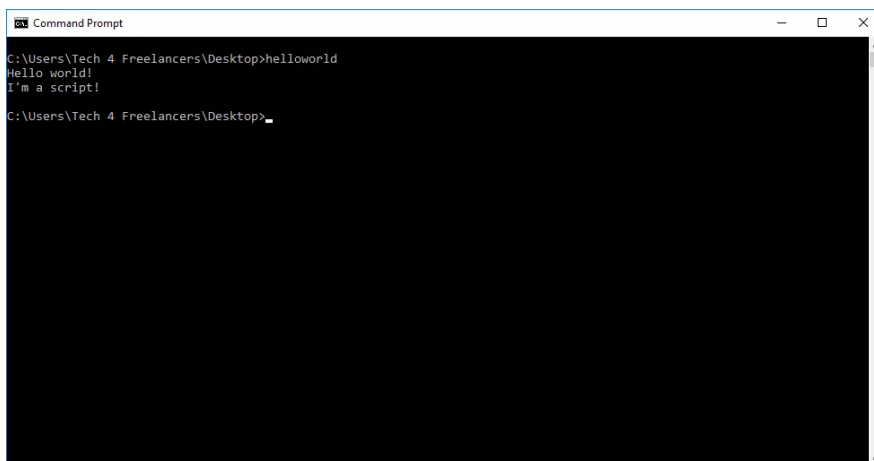
This looks much better, but notice how the command `echo off` is still displayed when we run the script. To prevent this, we will need to use `@echo off`.

Case #3: Using `@echo off`

Our code is identical to that in the previous example, save for a leading `@` at the beginning of the first line:

```
@echo off
echo Hello world!
echo I'm a script!
```

Running our code produces the following output:



```
C:\Users\Tech 4 Freelancers\Desktop>@echo off
C:\Users\Tech 4 Freelancers\Desktop>echo Hello world!
Hello world!
C:\Users\Tech 4 Freelancers\Desktop>echo I'm a script!
I'm a script!
C:\Users\Tech 4 Freelancers\Desktop>
```

This is the most desirable result of the three, if you ask me.

Using `@` to suppress a command's output

Of course, the usefulness of `@` doesn't end here. It's extremely useful when inside a `for` loop. Take this example:

