

CS50's Web Programming with Python and JavaScript_Lecture 0

April 9, 2023

Create a repository in Github.

Now, download this repository using `git clone url`

The screenshot shows a GitHub repository page for 'shammun / cs50WPPJ_Lecture0'. At the top, there is a large button with the text 'git clone <url>'. Below this, the repository name 'shammun / cs50WPPJ_Lecture0' is displayed along with a 'Unwatch' button. A navigation bar includes links for 'Code', 'Issues 0', 'Pull requests 0', 'Actions', 'Projects 0', 'Wiki', 'Security', and 'Insights'. A 'Quick setup — if you've done this kind of thing before' section is present, showing cloning options via 'Set up in Desktop' (disabled), 'HTTPS', and 'SSH' (selected) with the URL 'https://github.com/shammun/cs50WPPJ_Lecture0.git'. Below this, a note says 'Get started by creating a new file or uploading an existing file. We recommend every repository include a README, L'.

In **GitBash**, we use `git clone` in the following way.

```
User@WINDOWS-EKS4LMU MINGW64 ~
$ git clone https://github.com/shammun/cs50WPPJ_Lecture0.git
```

Go inside the directory `cs50WPPJ_Lecture0` and lists all the files which will be em pty.

```
User@WINDOWS-EKS4LMU MINGW64 ~
$ cd cs50WPPJ_Lecture0

User@WINDOWS-EKS4LMU MINGW64 ~/cs50WPPJ_Lecture0 (master)
$ ls

User@WINDOWS-EKS4LMU MINGW64 ~/cs50WPPJ_Lecture0 (master)
$ |
```

Now, add some files.

```
User@WINDOWS-EKS4LMU MINGW64 ~/cs50WPPJ_Lecture0 (master)
$ touch hello.html
```

0.0.1 git add

This tells that these are the files I want to commit or save a version.

```
User@WINDOWS-EKS4LMU MINGW64 ~/cs50WPPJ_Lecture0 (master)
$ git add hello.html
```

0.0.2 git commit

Take a snapshot of the repository in the current moment and save it.

```
git commit -m "message"
```

```
User@WINDOWS-EKS4LMU MINGW64 ~/cs50WPPJ_Lecture0 (master)
$ git commit -m "Added hello.html"
[master (root-commit) edb3d65] Added hello.html
 1 file changed, 9 insertions(+)
 create mode 100644 hello.html
```

0.0.3 git status

What's currently going on in the repository?

```
User@WINDOWS-EKS4LMU MINGW64 ~/cs50WPPJ_Lecture0 (master)
$ git status
On branch master
Your branch is based on 'origin/master', but the upstream is gone.
  (use "git branch --unset-upstream" to fixup)

nothing to commit, working tree clean

User@WINDOWS-EKS4LMU MINGW64 ~/cs50WPPJ_Lecture0 (master)
$
```

0.0.4 git push

Send those changes to Github

```
User@WINDOWS-EKS4LMU MINGW64 ~/cs50WPPJ_Lecture0 (master)
$ git push
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 302 bytes | 302.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/shammun/cs50WPPJ_Lecture0.git
 * [new branch]      master -> master

User@WINDOWS-EKS4LMU MINGW64 ~/cs50WPPJ_Lecture0 (master)
$
```

Now, we see that **hello.html** exists in Github.

Branch: master ▾ cs50WPPJ_Lecture0 / hello.html

Find file Copy path

shammun Added hello.html 12 minutes ago

1 contributor

9 lines (9 sloc) | 128 Bytes

Raw Blame History

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>My webpage!</title>
5   </head>
6   <body>
7     Hello, world!
8   </body>
9 </html>
```

0.0.5 git pull

Opposite of git push. What if someone else is also working on your repository and made some changes to it and pushed them to the repository. Now we want to download these latest changes.

Now, change code inside Github by clicking on the pen sign.

9 lines (9 sloc) | 128 Bytes

Raw Blame History

Edit this file

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>My webpage!</title>
5   </head>
6   <body>
7     Hello, world!
8   </body>
9 </html>
```

Now, add something inside **body** tag and then click on **Commit changes** at the bottom.

The screenshot shows a GitHub repository page for 'shammun / cs50WPPJ_Lecture0'. The 'Code' tab is selected. A file named 'hello.html' is open in the editor. The code content is:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>My webpage!</title>
5   </head>
6   <body>
7     Hello, world! Hello again
8   </body>
9 </html>
```

Below the editor, a 'Commit changes' dialog box is displayed. It includes fields for a commit message ('Update hello.html'), an optional extended description, and an email address ('sha_is13@yahoo.com'). It also shows options for committing directly to the 'master' branch or creating a new branch for a pull request. At the bottom are 'Commit changes' and 'Cancel' buttons.

Now, in the Git bash, pull these changes to put these changes in the computer.

```
User@WINDOWS-EKS4LMU MINGW64 ~/cs50WPPJ_Lecture0 (master)
$ git pull
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 681 bytes | 52.00 KiB/s, done.
From https://github.com/shammun/cs50WPPJ_Lecture0
  edb3d65..b12e870  master      -> origin/master
Updating edb3d65..b12e870
Fast-forward
 hello.html | 4 +---
 1 file changed, 2 insertions(+), 2 deletions(-)

User@WINDOWS-EKS4LMU MINGW64 ~/cs50WPPJ_Lecture0 (master)
$ |
```

Now, suppose we make some changes locally in the computer.

```
1  <!DOCTYPE html>
2  <html>
3  |  <head>
4  |  |  <title>My webpage!</title>
5  |  </head>
6  |  <body>
7  |  |  Hello, world! Hello again  Hello a third time
8  |  </body>
9  </html>
```

At the same time, someone else also make some changes in the Github.

shammun / cs50WPPJ_Lecture0

Code Issues 0 Pull requests 0 Actions Projects 0 Wiki Security Insights Settings

cs50WPPJ_Lecture0 / hello.html Cancel

Edit file Preview changes Spaces 3 No wrap

```
1  <!DOCTYPE html>
2  <html>
3  |  <head>
4  |  |  <title>My webpage!</title>
5  |  </head>
6  |  <body>
7  |  |  Hello, world! Hello again  Hello a third time
8  |  </body>
9  </html>
```

Now, if we want to pull code from the Github, there will be some conflicts.

0.0.6 Merge Conflicts

The terminal window shows two sets of code changes:

- your changes:**

```
a = 1
<<<< HEAD
{
b = 2
=====
{
b = 0
>>>> 57656c636f6d6520746f20576562
c = 3
d = 4
e = 5
```
- remote changes:**

```
a = 1
b = 2

c = 3
d = 4
e = 5
```

A vertical arrow labeled "conflicting commit" points from the bottom of the local changes section towards the bottom of the remote changes section.

Get rid of the lines that you don't want, the ««, the »» and the part of the code line that you don't want, remove whitespace and then commit.

The terminal window now displays the merged code, with the conflict markers removed:

```
a = 1
b = 2

c = 3
d = 4
e = 5
```

```
User@WINDOWS-EKS4LMU MINGW64 ~/cs50WPPJ_Lecture0 (master)
$ git pull
Auto-merging hello.html
CONFLICT (content): Merge conflict in hello.html
Automatic merge failed; fix conflicts and then commit the result.

User@WINDOWS-EKS4LMU MINGW64 ~/cs50WPPJ_Lecture0 (master |MERGING)
$ |
```

```
< hello.html >
C: > Users > User > cs50WPPJ_Lecture0 > < hello.html > html > body > ?
1  <!DOCTYPE html>
2  <html>
3  |   <head>
4  |     <title>My webpage!</title>
5  |   </head>
6  |   <body>
Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
7  |<<<<< HEAD (Current Change)
8  |   Hello, world! Hello again Hello a third time
9  =====
10 |   Hello, world! Hello again Hello Hello Hello
11 >>>> e69894ef556da3277436f6c3374744af4963f720 (Incoming Change)
12 |   </body>
13 </html>
14
```

Now, keep the line that you want and save.

Now, **git add** and **git commit** (and also **git push**).

```
User@WINDOWS-EKS4LMU MINGW64 ~/cs50WPPJ_Lecture0 (master |MERGING)
$ git add hello.html

User@WINDOWS-EKS4LMU MINGW64 ~/cs50WPPJ_Lecture0 (master |MERGING)
$ git commit -m "merge conflct"
[master 6727198] merge conflct

User@WINDOWS-EKS4LMU MINGW64 ~/cs50WPPJ_Lecture0 (master)
$ git push
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 429 bytes | 429.00 KiB/s, done.
Total 4 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To https://github.com/shamnum/cs50WPPJ_Lecture0.git
  e69894e..6727198  master -> master

User@WINDOWS-EKS4LMU MINGW64 ~/cs50WPPJ_Lecture0 (master)
$
```

We can also add and commit in a single command – **git commit -am “message”**

Now, github will take the version we have just committed by removing conflict.

The screenshot shows a GitHub repository page for 'shammun / cs50WPPJ_Lecture0'. The repository has 1 unwatched star, 0 forks, and 0 contributors. The 'Code' tab is selected, showing the file 'hello.html'. A merge conflict is indicated by the message 'shammun merge conflict' and a commit hash 'b23ca3e' from 6 minutes ago. The code editor displays the following HTML:

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>My webpage!</title>
5   </head>
6   <body>
7     Hello, world! Hello again Hello a third time
8   </body>
9 </html>
```

0.0.7 git log

Shows a history of all the commits that you have made.

```
User@WINDOWS-EKS4LMU MINGW64 ~/cs50WPPJ_Lecture0 (master)
$ git log
commit 67271982b90a269d6d3ea8b0eb4744c21ddb0bfc (HEAD -> master, origin/master)
Merge: b23ca3e e69894e
Author: shammun <sha_is13@yahoo.com>
Date:   Sat Apr  4 12:51:38 2020 +0600

    merge conflct

commit b23ca3ebf6369e49709d1fec865a10d5174973c3
Author: shammun <sha_is13@yahoo.com>
Date:   Sat Apr  4 12:47:34 2020 +0600

    merge conflct

commit e69894ef556da3277436f6c3374744af4963f720
Author: Shammunul Islam <sha_is13@yahoo.com>
Date:   Sat Apr  4 12:36:54 2020 +0600

    Update hello.html

commit b12e87034a2d12c6e9aea59e16eca97fbe550be2
Author: Shammunul Islam <sha_is13@yahoo.com>
Date:   Sat Apr  4 12:17:35 2020 +0600

....skipping...
commit 67271982b90a269d6d3ea8b0eb4744c21ddb0bfc (HEAD -> master, origin/master)
Merge: b23ca3e e69894e
Author: shammun <sha_is13@yahoo.com>
Date:   Sat Apr  4 12:51:38 2020 +0600

    merge conflct

commit b23ca3ebf6369e49709d1fec865a10d5174973c3
Author: shammun <sha_is13@yahoo.com>
Date:   Sat Apr  4 12:47:34 2020 +0600

    merge conflct

commit e69894ef556da3277436f6c3374744af4963f720
Author: Shammunul Islam <sha_is13@yahoo.com>
Date:   Sat Apr  4 12:36:54 2020 +0600

    Update hello.html

commit b12e87034a2d12c6e9aea59e16eca97fbe550be2
Author: Shammunul Islam <sha_is13@yahoo.com>
Date:   Sat Apr  4 12:17:35 2020 +0600

    Update hello.html

commit edb3d6585ee15b79d57c22866eff8f90c4decb05
Author: shammun <sha_is13@yahoo.com>
Date:   Sat Apr  4 12:00:55 2020 +0600

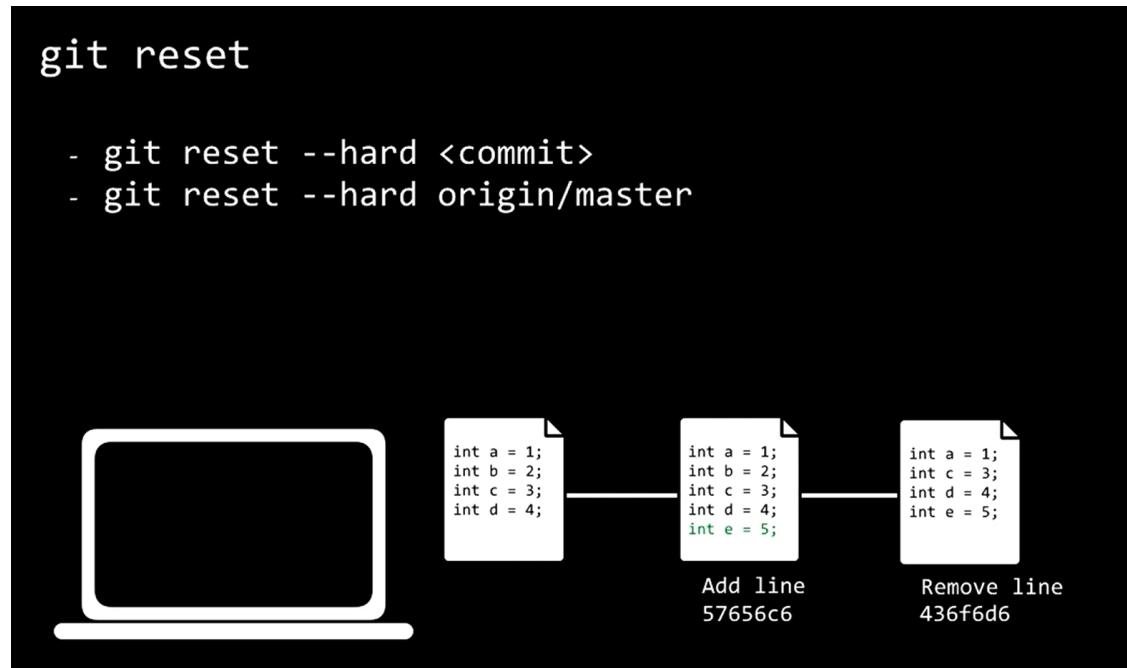
    Added hello.html
~
```

```
git log --oneline
```

See a brief log of all commits.

0.0.8 git reset

You ahve made some changes and then you realize that you messed up and need to go back to where things were before.



`git reset --hard (commit)` will reset the repository back to that version

`git reset --hard origin/master` If I have cloned my repository from Github then this command will set the version of the repository to the content that was originally in the Github.

Use he first couple of letters and numbers of a hash of a commit and then do the following.

```
User@WINDOWS-EKS4LMU MINGW64 ~/cs50WPPJ_Lecture0 (master)
$ git reset --hard edb3d6585e
HEAD is now at edb3d65 Added hello.html

User@WINDOWS-EKS4LMU MINGW64 ~/cs50WPPJ_Lecture0 (master)
$ |
```

Now, have a look at the `hello.html` in the desktop. We are back to the original version now.

```

↳ hello.html ✘

C: > Users > User > cs50WPPJ_Lecture0 > ↳ hello.html > ⚙ html

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>My webpage!</title>
5      </head>
6      <body>
7          Hello, world!
8      </body>
9  </html>

```

0.0.9 git reflog

Git keeps track of all the changes and so they are accessible.

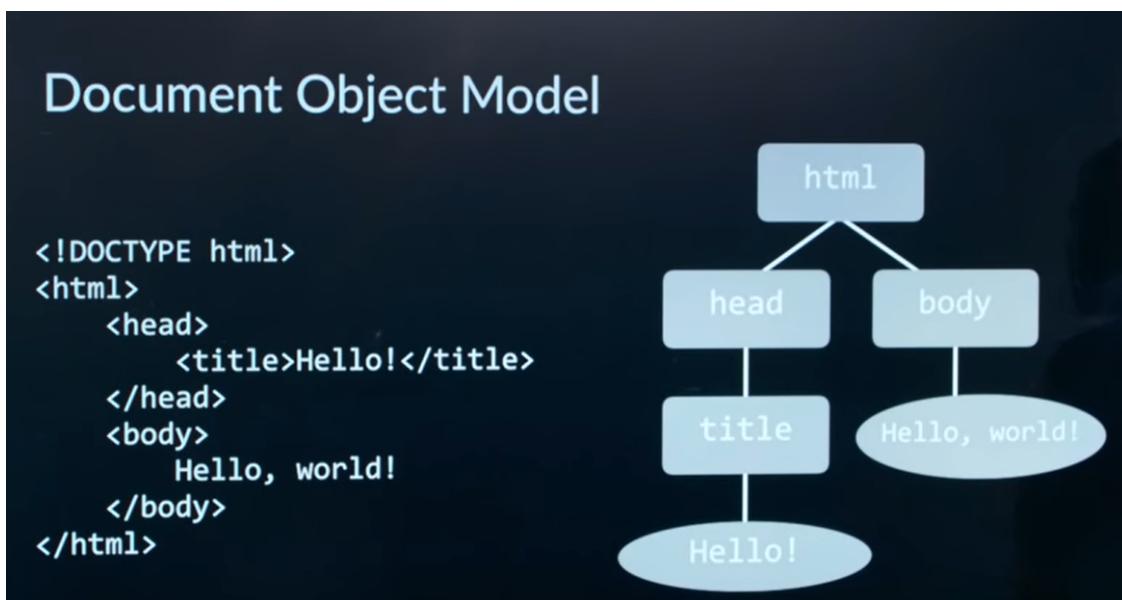
```

User@WINDOWS-EKS4LMU MINGW64 ~/cs50WPPJ_Lecture0 (master)
$ git reflog
edb3d65 (HEAD -> master) HEAD@{0}: reset: moving to edb3d6585e
6727198 (origin/master) HEAD@{1}: commit (merge): merge conflict
b23ca3e HEAD@{2}: commit: merge conflict
b12e870 HEAD@{3}: pull: Fast-forward
edb3d65 (HEAD -> master) HEAD@{4}: commit (initial): Added hello.html

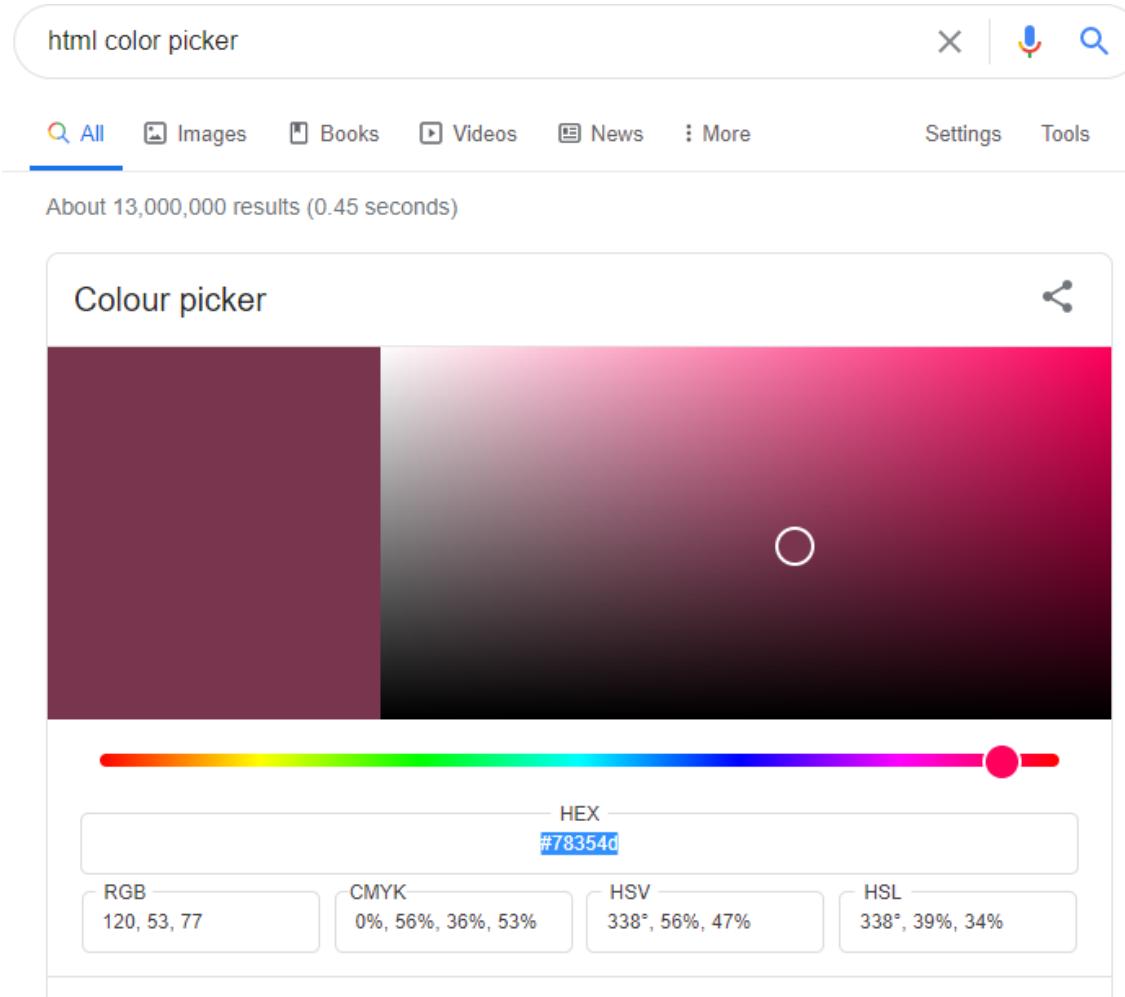
User@WINDOWS-EKS4LMU MINGW64 ~/cs50WPPJ_Lecture0 (master)

```

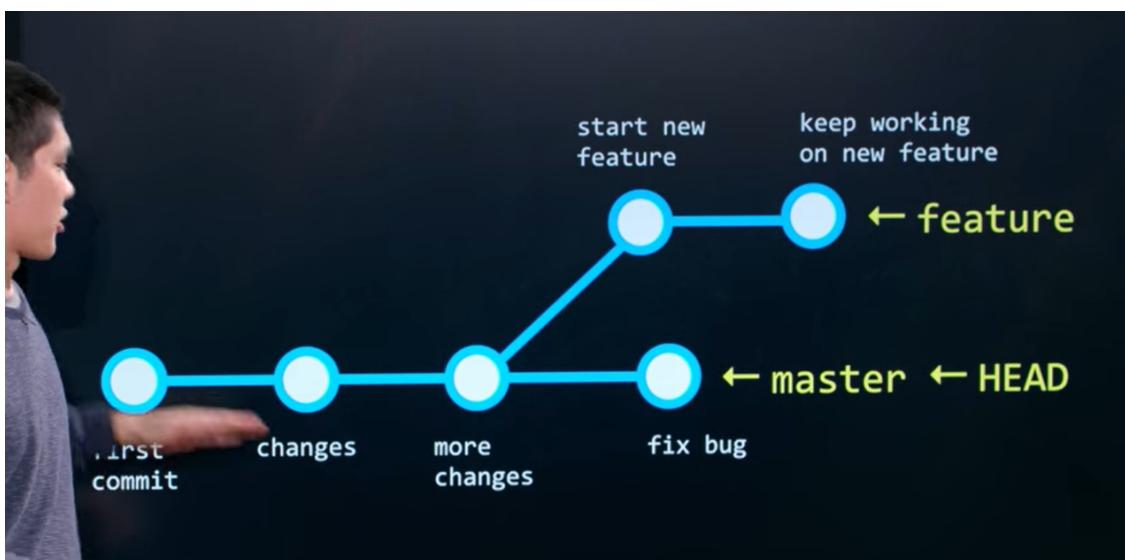
If I am working with multiple files, use `git add file_1 file_2 ...` or to add all files `git add *`

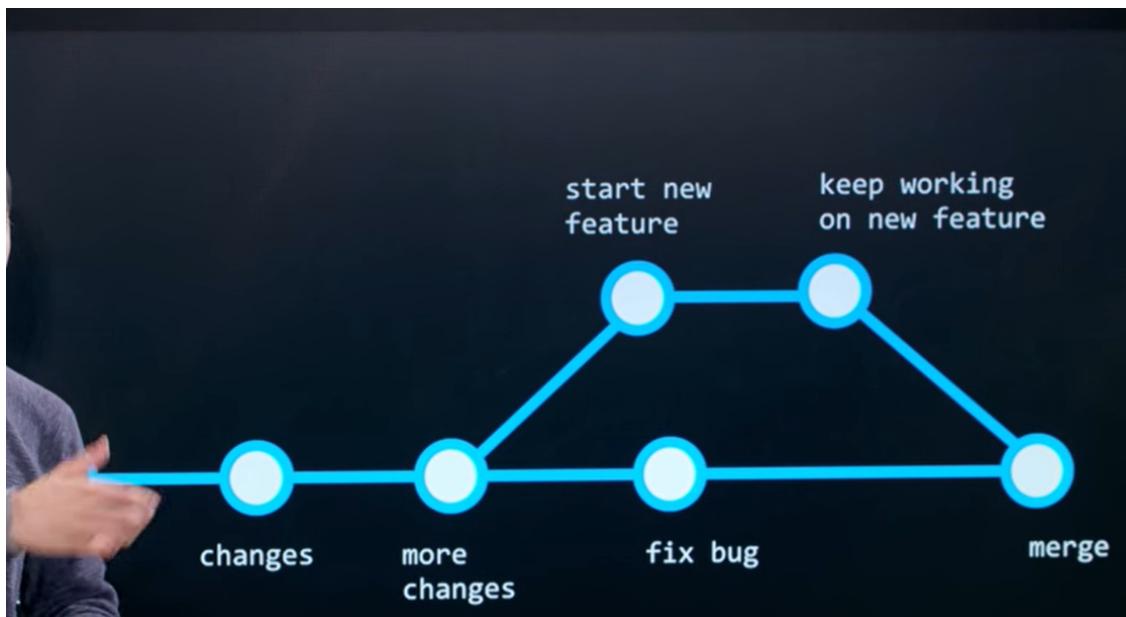
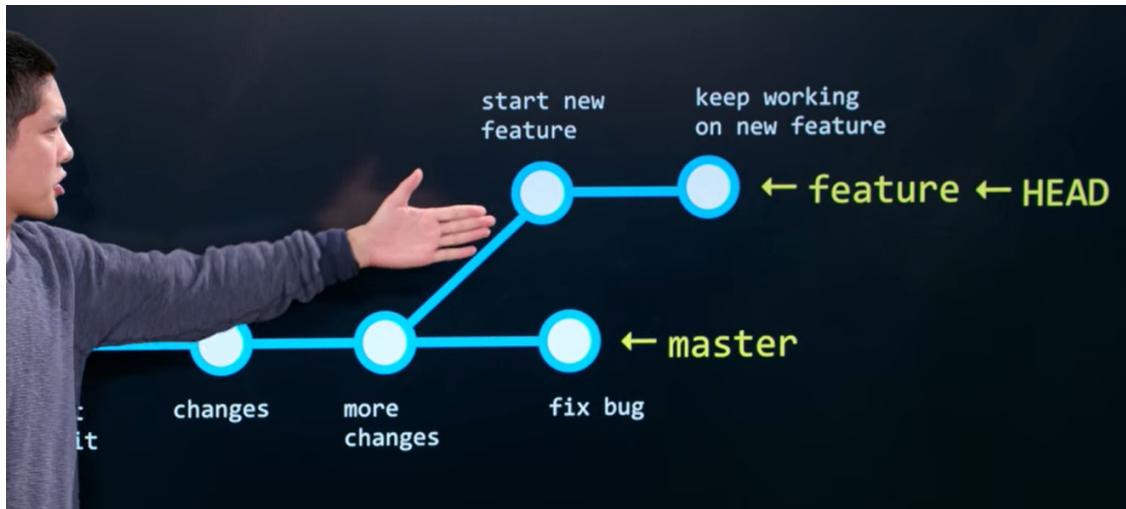


We can use color picker in Google by typing `html colour picker`



0.0.10 Branching





Create an HTML file, add it, commit it and then make a new branch **feature**

The screenshot shows the Visual Studio Code interface with the title bar "index.html - Visual Studio Code". The left sidebar lists files: hello.html, sabahat.html, div_span.html, index.html (which is the active tab), and table.html. The main editor area displays the following HTML code:

```
C: > Users > User > cs50WPPJ_Lecture0 > index.html > html > body
1  <!DOCTYPE html>
2  <html>
3  |   <head>
4  |   |       <title>Test!</title>
5  |   </head>
6  |   <body>
7  |       Hello, world!
8  |   </body>
9  </html>
```

git branch branch_name creates a new branch **branch_name**, name it anything you want.

```
User@WINDOWS-EKS4LMU MINGW64 ~
$ cd cs50WPPJ_Lecture0

User@WINDOWS-EKS4LMU MINGW64 ~/cs50WPPJ_Lecture0 (master)
$ git add index.html

User@WINDOWS-EKS4LMU MINGW64 ~/cs50WPPJ_Lecture0 (master)
$ got commit -m "Add index.html"
bash: got: command not found

User@WINDOWS-EKS4LMU MINGW64 ~/cs50WPPJ_Lecture0 (master)
$ git commit -m "Add index.html"
[master 9694992] Add index.html
 1 file changed, 9 insertions(+)
 create mode 100644 index.html

User@WINDOWS-EKS4LMU MINGW64 ~/cs50WPPJ_Lecture0 (master)
$ git branch
* master

User@WINDOWS-EKS4LMU MINGW64 ~/cs50WPPJ_Lecture0 (master)
$ git branch feature

User@WINDOWS-EKS4LMU MINGW64 ~/cs50WPPJ_Lecture0 (master)
$ git branch
  feature
* master

User@WINDOWS-EKS4LMU MINGW64 ~/cs50WPPJ_Lecture0 (master)
$ |
```

Right now, I am in the **master** branch where I am working on. To move to the feature branch, use **git checkout feature**

```
C:\Users\User\cs50WPPJ_Lecture1>git checkout feature
Switched to branch 'feature'

C:\Users\User\cs50WPPJ_Lecture1>git branch
* feature
  master
```

Now, make some changes in the **index.html** file

```
C: > Users > User > cs50WPPJ_Lecture1 > index.html > ...
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <title>My webpage!</title>
5      <style>
6        </style>
7    </head>
8    <body>
9      Hello, world!
10     Here is a new feature
11   </body>
12 </html>
```

Now, add this in the **feature** branch.

[]:

[]:

0.0.11 git checkout (commit_number)

Checkout the file from an older commit.

Suppose we have the following three commits and we want to go back to the second commit.

```
$ git log --oneline
155f427 (HEAD -> master) Third commit
7f99cbf Second commit
a42c9b9 First commit
```

Use **git checkout** followed by the commit number in the following way

If we want to go back to the second committed version of **index.html**

```
$ git checkout 7f99cbf index.html
Updated 0 paths from 7d31674
```

Now, **index.html** changes to the second version of commit.

Now, if we want to go back to the third version of commit

```
$ git reset HEAD index.html
```

```
$ git checkout -- index.html
```

Now, back to our branch example.

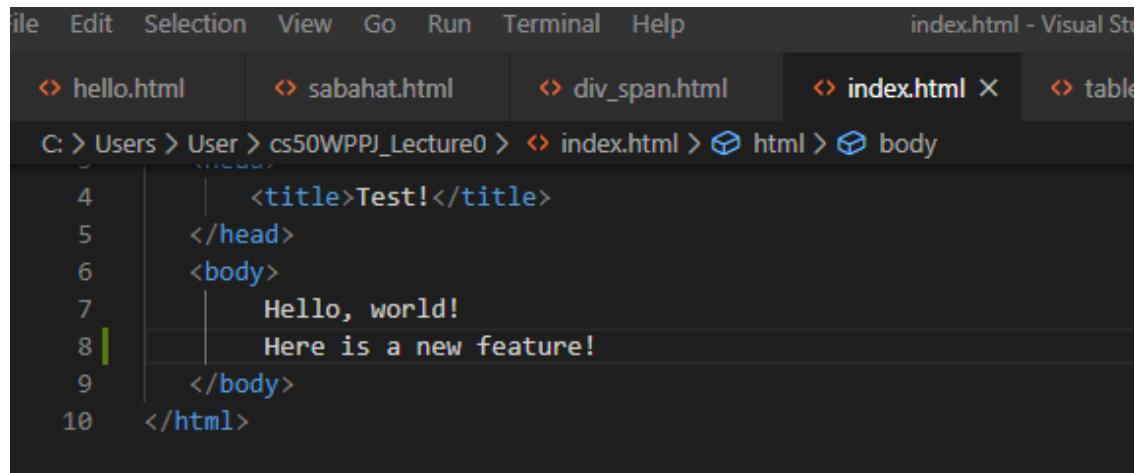
```
User@WINDOWS-EKS4LMU MINGW64 ~/cs50WPPJ_Lecture0 (master)
$ git checkout feature
Switched to branch 'feature'

User@WINDOWS-EKS4LMU MINGW64 ~/cs50WPPJ_Lecture0 (feature)
$ git branch
* feature
  master

User@WINDOWS-EKS4LMU MINGW64 ~/cs50WPPJ_Lecture0 (feature)
$ |
```

[]: If we want to go back to

Now, make some changes to the file **index.html**



```
File Edit Selection View Go Run Terminal Help index.html - Visual Studio Code
↳ hello.html ↳ sabahat.html ↳ div_span.html ↳ index.html X ↳ table.html
C: > Users > User > cs50WPPJ_Lecture0 > ↳ index.html > ↳ html > ↳ body
4 |   <title>Test!</title>
5 |   </head>
6 |   <body>
7 |     Hello, world!
8 |     Here is a new feature!
9 |   </body>
10  </html>
```

Now add and commit.

```
User@WINDOWS-EKS4LMU MINGW64 ~/cs50WPPJ_Lecture0 (feature)
$ git commit -am "Added another line"
[feature c9767d3] Added another line
 1 file changed, 1 insertion(+)
```

We still have this changed file.

The screenshot shows the Visual Studio Code interface with the file 'index.html' open. The code contains the following HTML:

```
C: > Users > User > cs50WPPJ_Lecture0 > < index.html > html > body
4 |     <title>Test!</title>
5 |     </head>
6 |     <body>
7 |         Hello, world!
8 |         Here is a new feature!
9 |     </body>
10    </html>
```

Now, let's check the **master** branch and we will see that **index.html** file is the old one.

```
User@WINDOWS-EKS4LMU MINGW64 ~/cs50WPPJ_Lecture0 (feature)
$ git branch
* feature
  master

User@WINDOWS-EKS4LMU MINGW64 ~/cs50WPPJ_Lecture0 (feature)
$ git checkout master
Switched to branch 'master'
Your branch and 'origin/master' have diverged,
and have 1 and 4 different commits each, respectively.
(use "git pull" to merge the remote branch into yours)
```

The screenshot shows the Visual Studio Code interface with the file 'index.html' open. The code contains the following HTML:

```
C: > Users > User > cs50WPPJ_Lecture0 > < index.html > html > body
1  <!DOCTYPE html>
2  <html>
3  |     <head>
4  |         <title>Test!</title>
5  |     </head>
6  |     <body>
7  |         Hello, world!
8  |     </body>
9  </html>
```

So the change is not in the master branch, only on the feature branch.

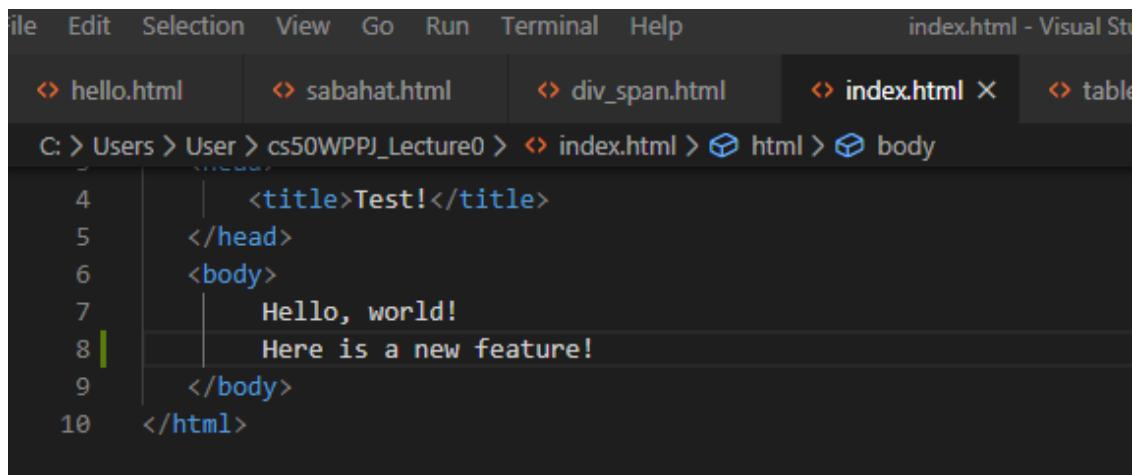
git checkout feature to get back to feature branch.

We can use **git commit** on each branch to see different changes.

Now, combine these two branches together. First, get inside the **master** branch and then use **git merge feature** to merge with **feature**.

```
User@WINDOWS-EKS4LMU MINGW64 ~/cs50WPPJ_Lecture0 (master)
$ git merge feature
Updating 9694992..c9767d3
Fast-forward
 index.html | 1 +
 1 file changed, 1 insertion(+)
```

Now, we will have a merged file.



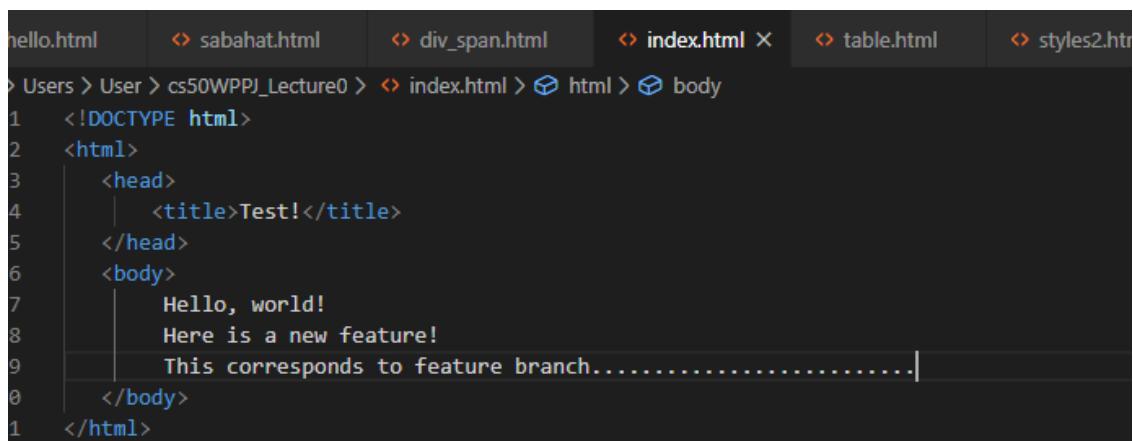
The screenshot shows the Visual Studio Code interface with the title bar "index.html - Visual Studio Code". Below the title bar, there are tabs for "hello.html", "sabahat.html", "div_span.html", "index.html", and "table.html". The "index.html" tab is active. The code editor displays the following HTML:

```
C: > Users > User > cs50WPPJ_Lecture0 > index.html > html > body
4     <title>Test!</title>
5     </head>
6     <body>
7         Hello, world!
8         Here is a new feature!
9     </body>
10    </html>
```

Now, switch to **feature** branch.

```
User@WINDOWS-EKS4LMU MINGW64 ~/cs50WPPJ_Lecture0 (master)
$ git checkout feature
Switched to branch 'feature'
```

Now, add a new line to the file **index.html**.



The screenshot shows the Visual Studio Code interface with the title bar "index.html - Visual Studio Code". Below the title bar, there are tabs for "hello.html", "sabahat.html", "div_span.html", "index.html", "table.html", and "styles2.css". The "index.html" tab is active. The code editor displays the following HTML:

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>Test!</title>
5      </head>
6      <body>
7          Hello, world!
8          Here is a new feature!
9          This corresponds to feature branch....|
```

Now, add and commit.

```
User@WINDOWS-EKS4LMU MINGW64 ~/cs50WPPJ_Lecture0 (feature)
$ git commit -am "Added a new line in the feature branch"
[feature ea49b36] Added a new line in the feature branch
 1 file changed, 1 insertion(+)
```

Now, try to push this and we will get an error as **Github** doesn't have any branch.

```
User@WINDOWS-EKS4LMU MINGW64 ~/cs50WPPJ_Lecture0 (feature)
$ git push
fatal: The current branch feature has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin feature
```

To resolve this, push to a new branch in Github.

Basic Git Commands

- **git reset <file>**
 - unstage a staged file, but leave working directory unchanged
- **git reset**
 - reset the staging area to the last commit without disturbing the working directory

0.0.12 git push --set-upstream origin feature

```
User@WINDOWS-EKS4LMU MINGW64 ~/cs50WPPJ_Lecture0 (feature)
$ git push --set-upstream origin feature
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 4 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (9/9), 954 bytes | 954.00 KiB/s, done.
Total 9 (delta 2), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (2/2), done.
remote:
remote: Create a pull request for 'feature' on GitHub by visiting:
remote:     https://github.com/shammun/cs50WPPJ_Lecture0/pull/new/feature
remote:
To https://github.com/shammun/cs50WPPJ_Lecture0.git
 * [new branch]      feature -> feature
Branch 'feature' set up to track remote branch 'feature' from 'origin'.
```

Now, in the Github, we see that we have the **feature** branch added.

shammun / cs50WPPJ_Lecture0

Code Issues 0 Pull requests 0 Actions Projects 0 Wiki Security Insights

No description, website, or topics provided.

Manage topics

4 commits 2 branches 0 packages 0 releases 1 environn

Your recently pushed branches:

feature (1 minute ago)

Branch: feature ▾ New pull request Create new file Upload files

This branch is 3 commits ahead, 4 commits behind master.

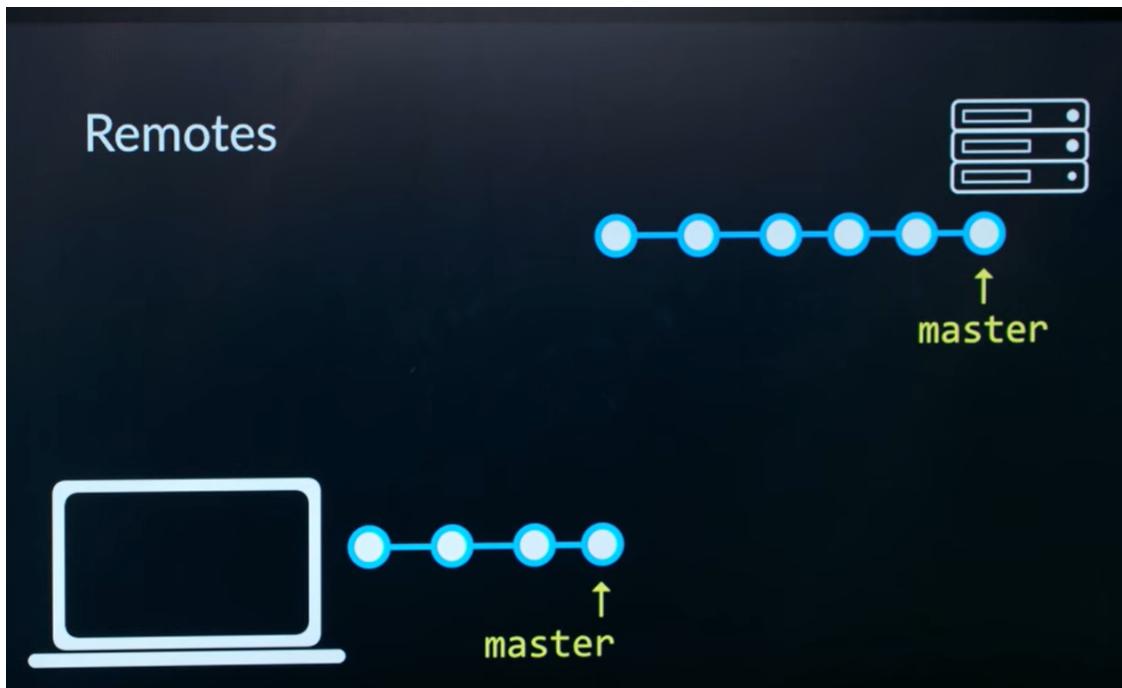
shammun Added a new line in the feature branch Lat

hello.html Added hello.html

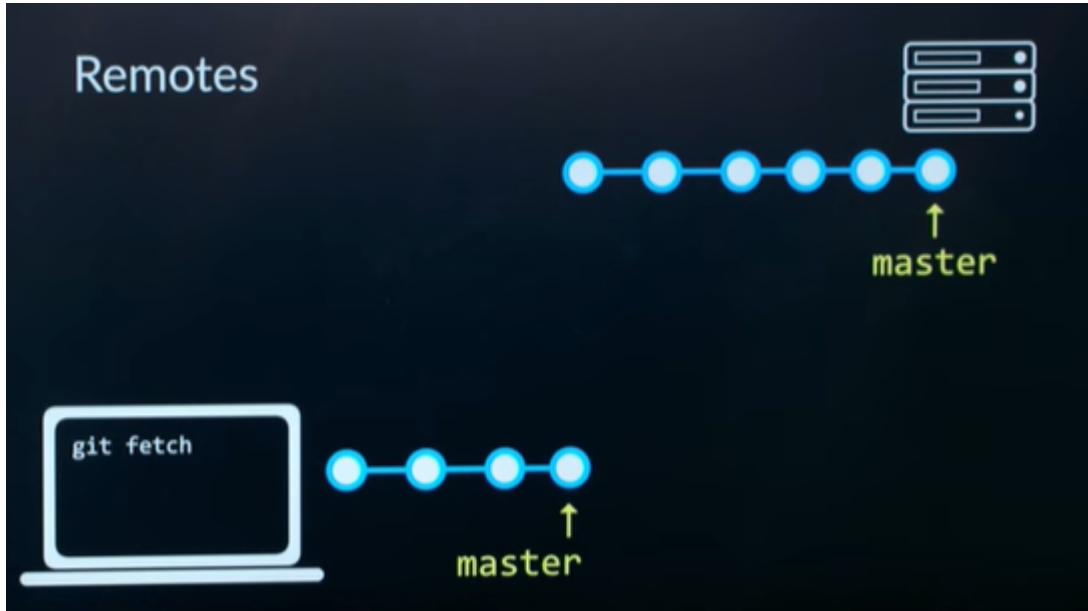
index.html Added a new line in the feature branch

0.0.13 Remotes

We can have a version of the repository that's stored somewhere else on the computer, or stored somewhere else on the internet, in particular, on Github. The name of the version of the repository that lives somewhere else is called remote.

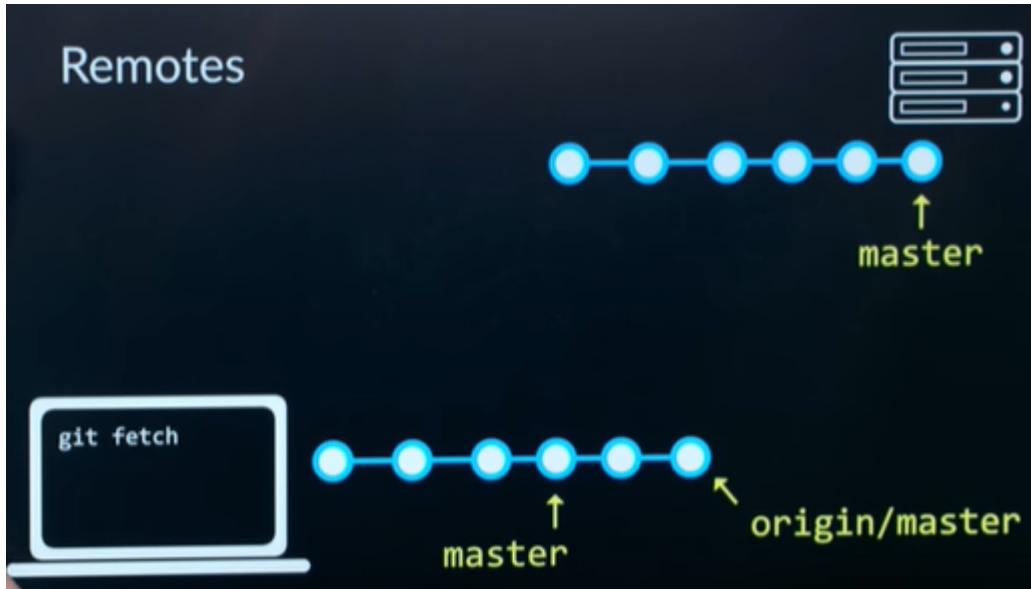


If I want to work with the remote version, the version in the Github, use `git fetch`



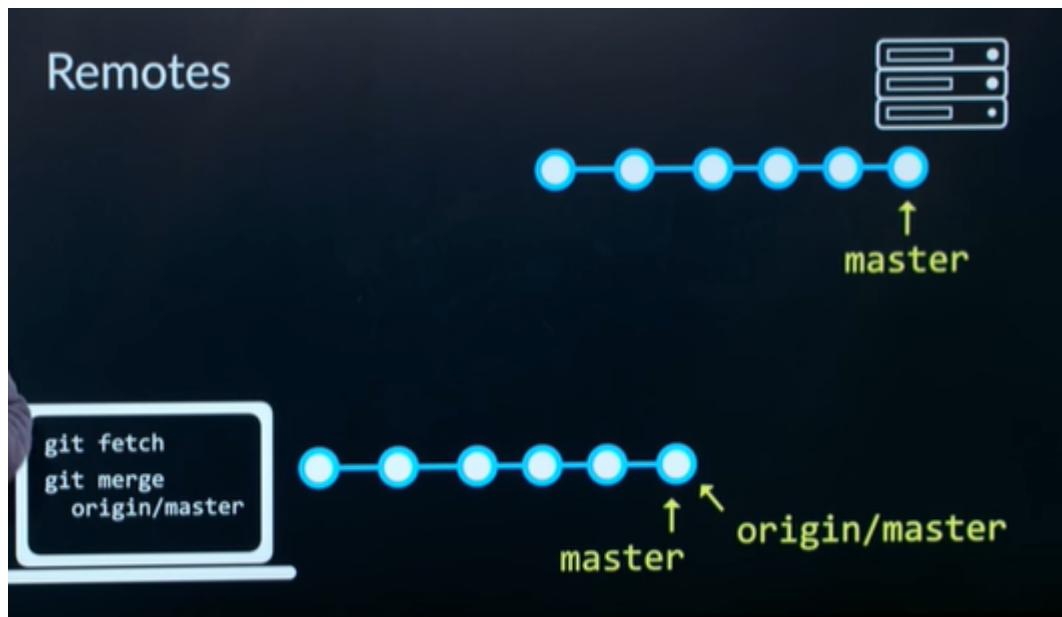
0.0.14 git fetch

After, we execute `git fetch`, we have `origin/master` branch from remote.



Now, I want my `master` branch to be updated to reflect these changes. I want to merge where I am with where `origin/master` is.

Use `git merge origin/master`



These commands can be simplified just to **git pull**

0.0.15 Forks

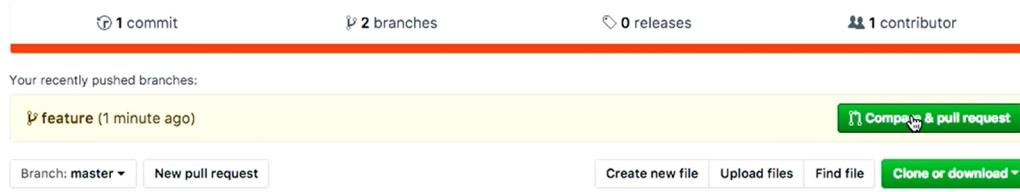
Fork of a repository is just an entirely separate version of that repository that get copied based off an original. Now, we can add branches to that repository, we can push it and we can pull from it... This will not affect the original version of the repository.

After someone forks an opensource project, makes some changes to it and when he wants his changes to be merged back minto the original code, he submits a pull request.

0.0.16 Pull requests

You would like for your changes to be brought in from one branch of some repository to some other branch of the same repository or even a differnt repository.

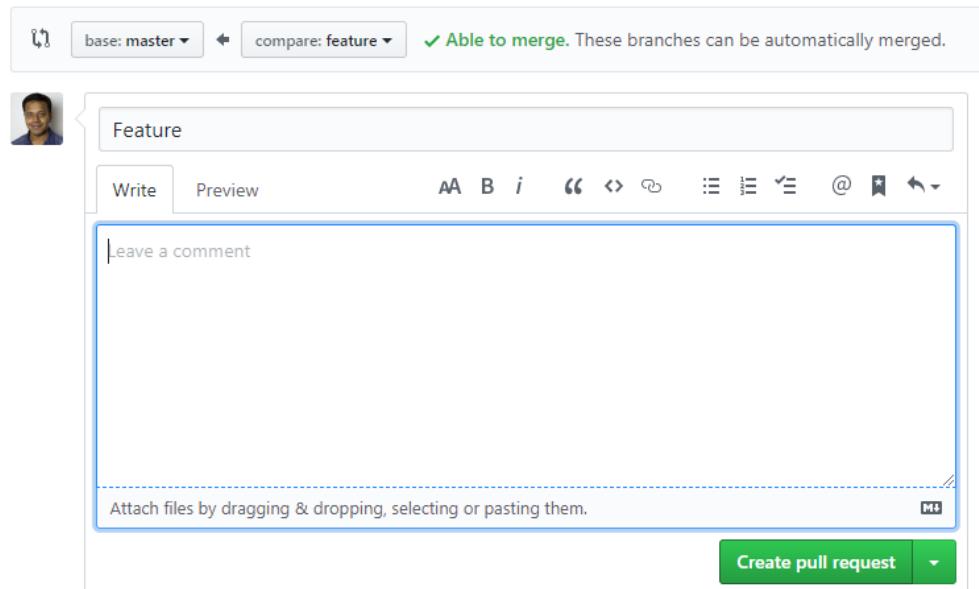
Click on **Compare & pull request**



Now, click on **Create pul request**

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).



0.0.17 git init

Initialize a repository using `git init`.

```
User@WINDOWS-EKS4LMU MINGW64 ~/cs50WPPJ_Lecture1
$ git init
Initialized empty Git repository in C:/Users/User/cs50WPPJ_Lecture1/.git/
```

0.0.18 Create a new Github repository and link the local rep with this Github repository.

```
User@WINDOWS-EKS4LMU MINGW64 ~/cs50WPPJ_Lecture1 (master)
$ git remote add origin https://github.com/shammun/cs50WPPJ_Lecture1.git
```

Now, create some files `links0.html`, `hello.html`, ... in the local repository. Now add these files and commit.

```
User@WINDOWS-EKS4LMU MINGW64 ~/cs50WPPJ_Lecture1 (master)
$ git add .

User@WINDOWS-EKS4LMU MINGW64 ~/cs50WPPJ_Lecture1 (master)
$ git commit -m "first commit"
[master (root-commit) 4c4ec66] first commit
 3 files changed, 57 insertions(+)
 create mode 100644 hello.html
 create mode 100644 links0.html
 create mode 100644 links1.html
```

Now, push these changes using `git push -u origin master`

```
User@WINDOWS-EKS4LMU MINGW64 ~/cs50WPPJ_Lecture1 (master)
$ git push -u origin master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 864 bytes | 864.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/shammun/cs50WPPJ_Lecture1.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
```

0.0.19 Uploading an entire folder

```
git push -u -f origin master
```

```
git init
git add -A
git commit -m 'Added my project'
git remote add origin git@github.com:scotch-io/my-new-project.git
git push -u -f origin master
```

HTML5 Organization

- <header>
- <nav>
- <section>
- <footer>
- ...

New HTML5 Elements

- <audio>
- <video>
- <datalist>
- ...

0.0.20 Combinator

0.0.21 selector selector

Matches elements that are descendants of another element

```
Lecture0    sabahat.html    div_span.html    index.html    descendant.html x
Users > User > cs50WPPJ_Lecture1 > descendant.html > html > body
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>Test!</title>
5          <style>
6              ol li {
7                  color: red;
8              }
9          </style>
0      </head>
1      <body>
2          <ol>
3              <li>list item one</li>
4              <li>list item two</li>
5                  <ul>
6                      <li>sublist item one</li>
7                      <li>sublist item two</li>
8                  </ul>
9                  <li>list item three</li>
0          </ol>
1      </body>
2  </html>
```



```
1. list item one
2. list item two
    o sublist item one
    o sublist item two
3. list item three
```

0.0.22 selector > selector Child combinator

Matches elements that are children of another element

```
C: > Users > User > cs50WPPJ_Lecture1 > descendant2.html > ...
```

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>Test!</title>
5          <style>
6              ol > li {
7                  color: red;
8              }
9          </style>
10     </head>
11     <body>
12         <ol>
13             <li>list item one</li>
14             <li>list item two</li>
15             <ul>
16                 <li>sublist item one</li>
17                 <li>sublist item two</li>
18             </ul>
19             <li>list item three</li>
20         </ol>
21     </body>
22 </html>
```

Only selects the immediate children.

1. list item one
2. list item two
 - o sublist item one
 - o sublist item two
3. list item three

0.0.23 Attribute selectors

```
div_span.html index.html descendant2.html links0.html links1.html attribute.html
users > User > cs50WPPJ_Lecture1 > attribute.html > html > body > form > div
<!DOCTYPE html>
<html>
  <head>
    <title>My webpage!</title>
    <style>
      input[type=text] {
        background-color: red;
      }

      input[type=number] {
        background-color: yellow;
      }
    </style>
  </head>
  <body>
    <form>
      <div>
        <input type="text" placeholder = "First name" name = "name">
        <input type="text" placeholder = "Last name" name = "name">
        <input name = "name" type = "number" placeholder = "Age">
      </div>

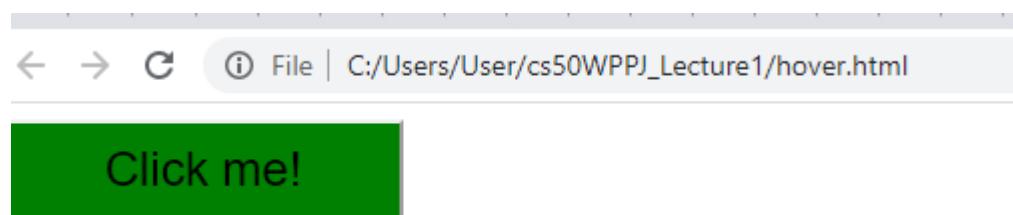
      <div>
        Favorite color?
        <input name="color" type="radio" value="red">Red
        <input name="color" type="radio" value="green">Red
        <input name="color" type="radio" value="blue">Red
        <input name="color" type="radio" value="other">Red
      </div>
    </form>
  </body>
</html>
```

First name Last name Age

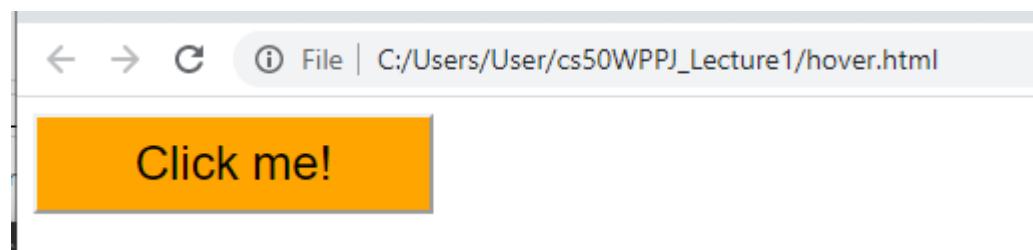
Favorite color? Red Red Red Red

0.0.24 pseudo-class

```
index.html descendant2.html links0.html links1.html attribute.html hover.html X
Users > User > cs50WPPJ_Lecture1 > hover.html > html > body > button
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>Test!</title>
5          <style>
6              button {
7                  width: 200px;
8                  height: 50px;
9                  font-size: 24px;
10                 background-color: green;
11             }
12
13             button:hover {
14                 background-color: orange;
15             }
16         </style>
17     </head>
18     <body>
19         <button>Click me!</button>
20     </body>
21 </html>
```



But, when we hover the button



0.0.25 Pseudo element

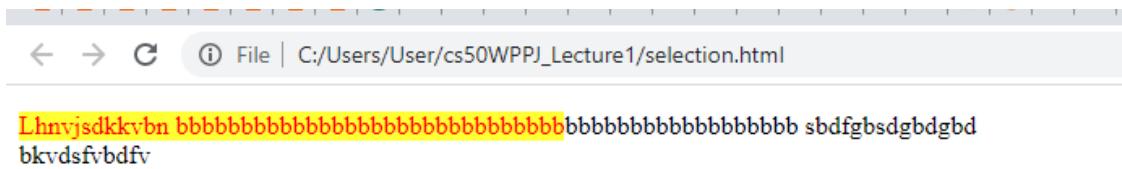
```
ex.html    <-- descendant2.html    <-- links0.html    <-- links1.html    <-- attribute.html    <-- before.html >  
s > User > cs50WPPJ_Lecture1 > before.html > html  
<!DOCTYPE html>  
<html>  
  <head>  
    <title>Test!</title>  
    <style>  
      a::before {  
        content: "\21d2 Click here: ";  
        font-weight: bold;  
      }  
      button:hover {  
        background-color: orange;  
      }  
    </style>  
  </head>  
  <body>  
    <ul>  
      <li><a href="#">one link</a></li>  
      <li><a href="#">second link</a></li>  
      <li><a href="#">third link</a></li>  
    </ul>  
  </body>  
</html>
```

← → ⌂ ⓘ File | C:/Users/User/cs50WPPJ_Lecture1/before.html

- [⇒Click here: one link](#)
- [⇒Click here: second link](#)
- [⇒Click here: third link](#)

```
index.html descendant2.html links0.html links1.html attribute.html selection.html X
C: > Users > User > cs50WPPJ_Lecture1 > selection.html > html
1  <!DOCTYPE html>
2  <html>
3  |   <head>
4  |       <title>Test!</title>
5  |       <style>
6  |           p::selection {
7  |               color: red;
8  |               background-color: yellow;
9  |           }
10 |       </style>
11 |   </head>
12 |   <body>
13 |       <p>
14 |           Lhnvjsdkkvbn bbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
15 |
16 |           sbdfgbsdgbdgbd<br>
17 |
18 |           bkvdsvbdfv<br><br><br><br><br><br>
19 |
20 |           sdkjvndsfblkvbmdfbk bm;
21 |       </p>
22 |   </body>
23 |</html>
```

Now, when I highlight text, we get red colored font with yellow background.



sdkjvndsfblkvbmdfbk bm;

CSS Selectors

a, b	Multiple Element Selector
a b	Descendant Selector
a > b	Child Selector
a + b	Adjacent Sibling Selector
[a=b]	Attribute Selector
a:b	Pseudoclass Selector
a::b	Pseudoelement Selector

Basic selectors

Selector	Description	Example
element	Type selector. Matches an element.	p { color: red; } /* matches paragraphs */
.class	Class selector. Matches the value of a <code>class</code> attribute.	.warning { color: red; } /* matches elements containing class="warning" */
#id	ID selector. Matches the value of an <code>id</code> attribute.	#warning { color: red; } /* matches elements containing id="warning" */
*	Universal selector. Matches everything.	* { color: red; } /* matches everything */

Attribute selectors

Selector	Description	Example
[attribute]	Matches elements containing a given attribute.	a[href] { color: red; } /* matches a elements with an href attribute */
[attribute="x"]	Matches elements containing a given attribute with a given value.	a[href="/sitemap/] { color: red; } /* matches a elements with the attribute and value href="/sitemap/" */
[attribute~= "x"]	Matches elements containing a given attribute with a value that contains a sub-value within a space-separated list.	abbr[title="Style"] { color: red; } /* matches abbr elements with a title that contains 'Style' (such as in title="Cascading Style Sheets") */
[attribute = "x"]	Matches elements containing a given attribute with a value that contains a sub-value within a hyphen-separated list.	html[lang = "en"] { color: red; } /* matches html elements with a lang attribute that contains 'en' (such as in lang="en-gb") */
[attribute^= "x"]	Matches elements containing a given attribute with a value that starts with something.	a[href^="http://"] { color: red; } /* matches a elements with an href attribute, the value of which begins with 'http://' */

<code>[attribute\$="x"]</code>	Matches elements containing a given attribute with a value that ends with something.	<code>a[href\$=".com"] { color: red }</code> /* matches a elements with an href attribute, the value of which ends with '.com' */
<code>[attribute*="x"]</code>	Matches elements containing a given attribute with a value that contains something.	<code>a[href*="htmldog"] { color: red }</code> /* matches a elements with an href attribute, the value of which contains 'htmldog' */

ADVERTISEMENT

**Link To Us!**

If you've found HTML Dog useful, please consider linking to us.

Pseudo-classes

Selector	Description	Example
<code>:link</code>	Matches a link that has not been visited .	<code>a:link { color: blue }</code>
<code>:visited</code>	Matches a link that has been visited .	<code>a:visited { color: purple }</code>
<code>:active</code>	Matches an element that is being activated , such as a link being clicked on.	<code>a:active { color: red }</code>
<code>:hover</code>	Matches an element whose box is being hovered over by a cursor.	<code>a:hover { text-decoration: none }</code>
<code>:focus</code>	Matches an element that has focus , such as one that has been tabbed to.	<code>a:focus { border: 1px solid yellow }</code>
<code>:target</code>	Matches an element that has been linked to (via <code>...</code> , for example).	<code>h2:target { color: red }</code> /* matches a second-level heading that has been linked to */
<code>:lang()</code>	Matches an element of a given language .	<code>p:lang(fr) { color: red }</code> /* matches paragraphs that are declared, or otherwise considered, as French */
<code>:first-child</code>	Matches the first child of an element.	<code>p:first-child { color: red }</code> /* matches the first child, if it is a paragraph, of an element */
<code>:last-child</code>	Matches the last child of an element.	<code>div p:last-child { color: blue }</code> /* matches the last child, if it is a paragraph, of an element */
<code>:first-of-type</code>	Matches the first sibling of its type in an element.	<code>li:first-of-type { color: red }</code> /* matches the first instance of a list item inside an element */
<code>:last-of-type</code>	Matches the last sibling of its type in an element.	<code>li:last-of-type { color: blue }</code> /* matches the last instance of a list item inside an element */
<code>:nth-child()</code>	Matches an element that is the ordinal number child of its parent.	<code>p:nth-child(3) { color: red }</code> /* matches the third child, if it is a paragraph, of an element */
<code>:nth-last-child()</code>	Matches an element that is the ordinal number child, in reverse order of its parent.	<code>p:nth-last-child(2) { color: blue }</code> /* matches the next-to-last child, if it is a paragraph, of an element */

<code>:nth-of-type()</code>	Matches an element that is the ordinal number sibling of its type .	<code>li:nth-of-type(5) { color: red }</code> /* matches the fifth instance of a list item inside an element */
<code>:nth-last-of-type()</code>	Matches an element that is the ordinal number sibling, in reverse order, of its type .	<code>li:nth-last-of-type(5) { color: red }</code> /* matches the next-to-last instance of a list item inside an element */
<code>:only-child</code>	Matches an element if it is the only child of its parent.	<code>article p:only-child { color: red }</code> /* matches a paragraph if it is the only child of an article element */
<code>:only-of-type</code>	Matches an element if it is the only sibling of its type .	<code>article aside:only-of-type { color: blue }</code> /* matches an aside element if it is the only aside element in an article element */
<code>:empty</code>	Matches an element with no children, or content .	<code>td:empty { border-color: red }</code> /* matches table data cells with nothing in 'em */
<code>:root</code>	Matches the root element of a document. This will be the <code>html</code> element in HTML.	<code>:root { background: yellow }</code>
<code>:enabled</code>	Matches form control elements that are not disabled .	<code>input:enabled { border-color: lime }</code> /* matches input elements that are not disabled */
<code>:disabled</code>	Matches form control elements that are disabled .	<code>input:disabled { border-color: red }</code> /* matches input elements that are disabled */
<code>:checked</code>	Matches a radio or checkbox type input element that is checked .	<code>input:checked { outline: 3px solid yellow }</code> /* matches checked input elements */
<code>:not()</code>	Negotiation pseudo-class. Matches an element that does not match a selector.	<code>p:not(:first-child) { color: orange }</code> /* matches paragraphs that are not first children */

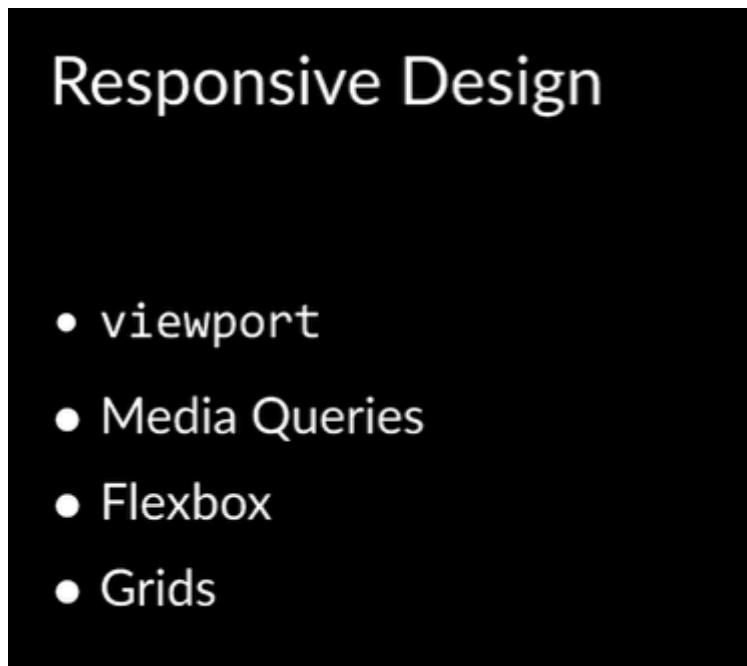
Pseudo-elements

Selector	Description	Example
<code>::first-line</code>	Matches the first textual line in an element.	<code>p::first-line { font-weight: bold }</code> /* matches the first line in a paragraph */
<code>::first-letter</code>	Matches the first letter in an element.	<code>p::first-letter { font-size: 2em }</code> /* matches the first letter in a paragraph */
<code>::before</code>	Used with the <code>content</code> property to generate content before the initial content of an element.	<code>h1::before { content: "*" }</code> /* places an asterisk at the start of a top-level heading */
<code>::after</code>	Used with the <code>content</code> property to generate content after the initial content of an element.	<code>h1::after { content: "+" }</code> /* places a plus-sign at the end of a top-level heading */

Combinators

Selector	Description	Example
<code>selector selector</code>	Descendant combinator. Matches elements that are descendants of another element.	<code>aside p { color: red }</code> /* matches paragraphs inside elements containing class="warning" */
<code>selector > selector</code>	Child combinator. Matches elements that are children of another element.	<code>.warning > p { color: red }</code> /* matches paragraphs that are children of elements containing class="warning" */
<code>selector + selector</code>	Adjacent sibling combinator. Matches elements that immediately follow another element.	<code>h1 + * { color: red }</code> /* matches the first element to follow a top-level heading */
<code>selector ~ selector</code>	General sibling combinator. Matches elements that follow another element.	<code>h2 ~ p { color: red }</code> /* matches every paragraph that follows a second-level heading */

0.0.26 Responsive design



```
C: > Users > User > cs50WPPJ_Lecture1 > print.html > ...
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <title>Test!</title>
5      <style>
6        @media print {
7          .screen-only {
8            display: none;
9          }
10       }
11     </style>
12   </head>
13   <body>
14     <p>This is a paragraph </p>
15     <p>This is another paragraph </p>
16     <p class = "screen-only">This paragraph won't appear you print t
17   </body>
18 </html>
```

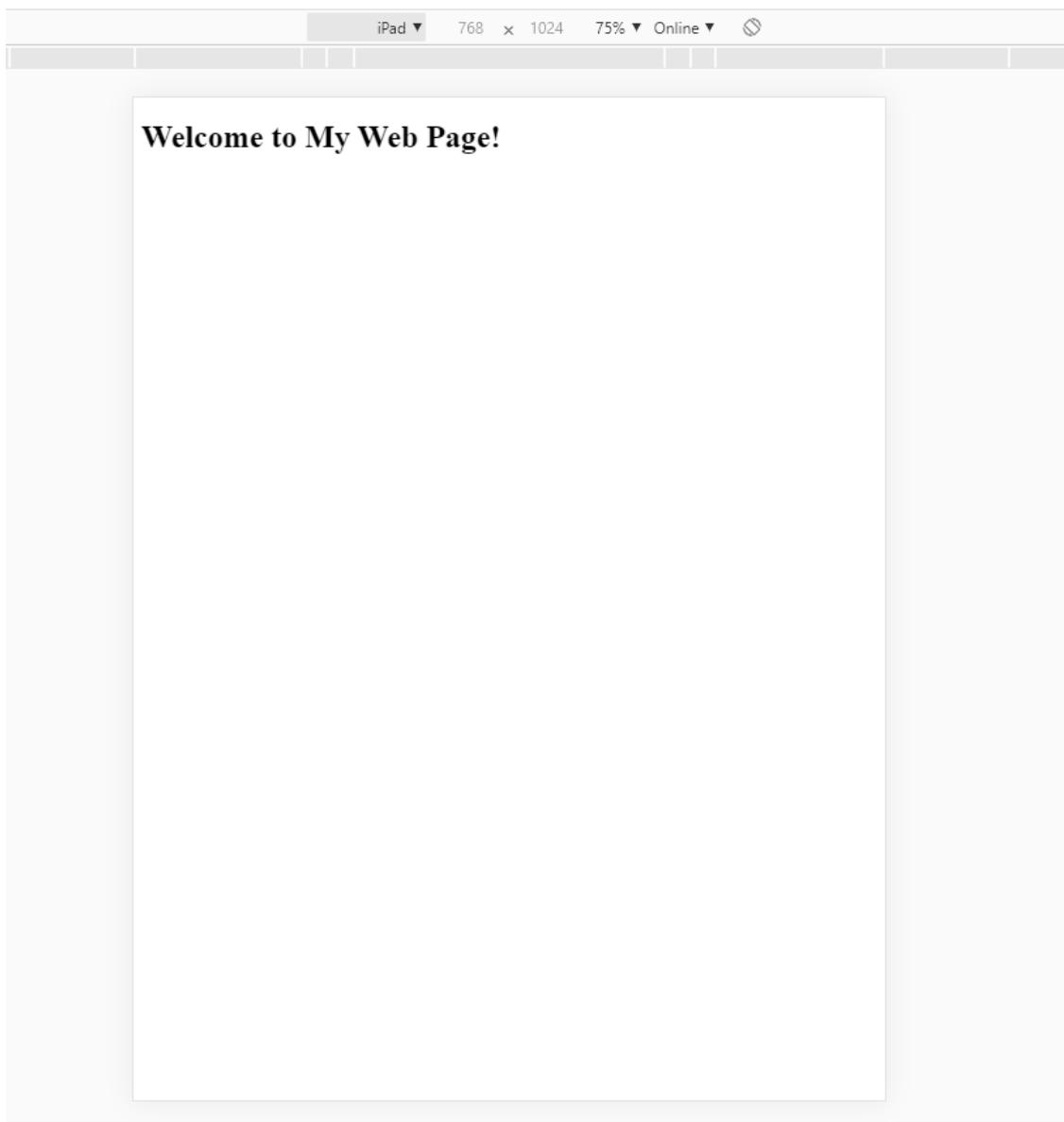
Don't print a particular line when printed

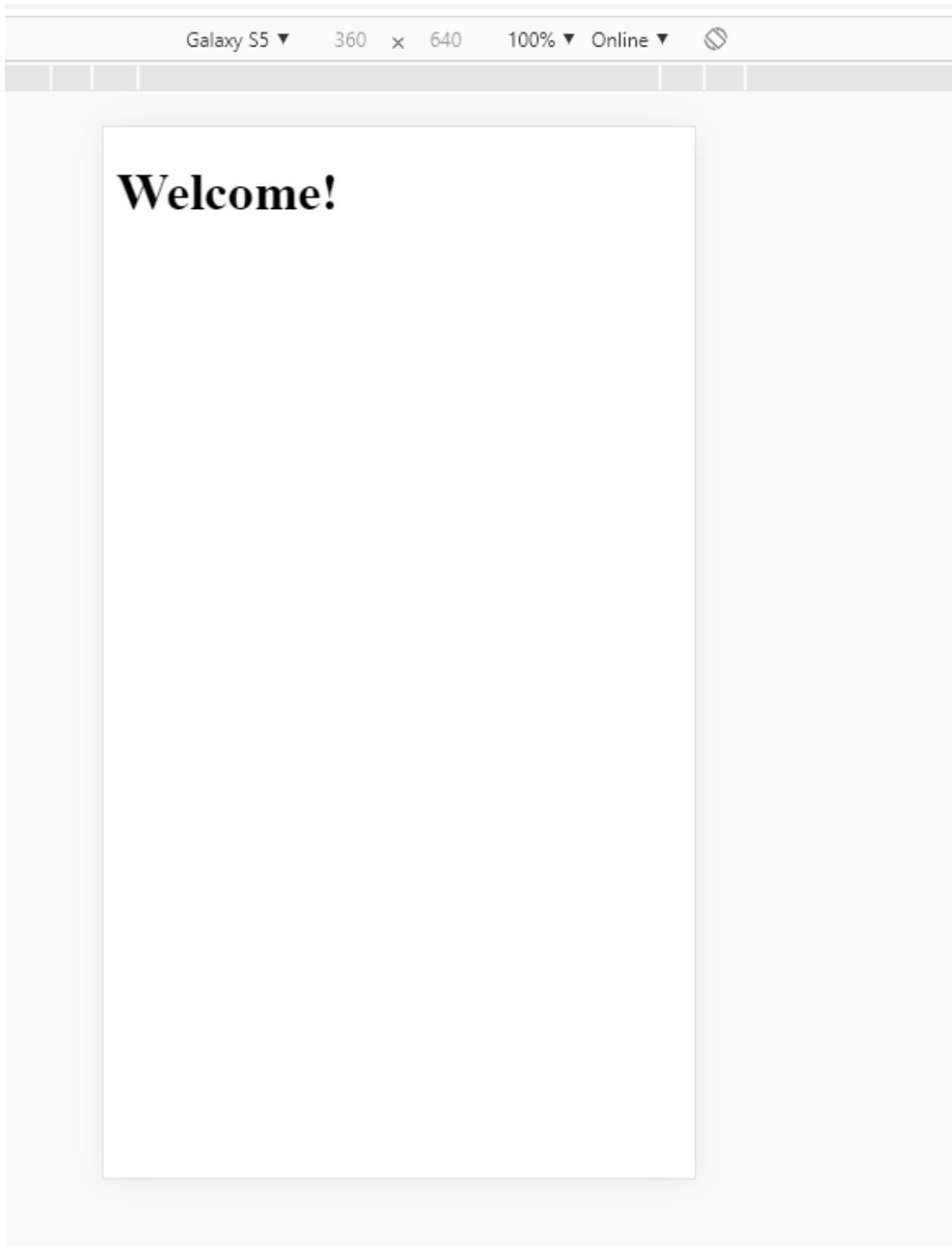
0.0.27 Responsive design

```
C: > Users > User > cs50WPPJ_Lecture1 > responsive0.html > ...
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <title>My Web Page!</title>
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <style>
7        @media (min-width: 500px) {
8          body {
9            background-color: red;
10         }
11       }
12
13        @media (max-width: 499px) {
14          body {
15            background-color: blue;
16          }
17        }
18      </style>
19    </head>
20    <body>
21      <h1>Welcome to My Web Page!</h1>
22    </body>
23 </html>
24
```

For the following example, text changes with different size of the viewport.

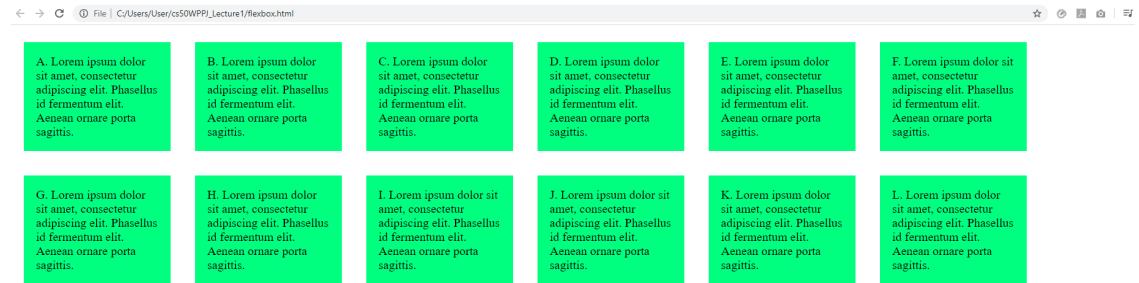
```
C:\> Users > User > cs50WPPJ_Lecture1 > responsive1.html > ...
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>My Web Page!</title>
5          <meta name="viewport" content="width=device-width, initial-scale=1.0">
6          <style>
7              @media (min-width: 700px) {
8                  h1::before {
9                      content: "Welcome to My Web Page!";
10                 }
11             }
13             @media (max-width: 699px) {
14                 h1::before {
15                     content: "Welcome!";
16                 }
17             }
18         </style>
19     </head>
20     <body>
21         <h1></h1>
22     </body>
23 </html>
```





0.0.28 Flexbox

```
4     <title>My Web Page!</title>
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">
6     <style>
7         .container {
8             display: flex;
9             flex-wrap: wrap;
10        }
11
12        .container > div {
13            background-color: springgreen;
14            font-size: 20px;
15            margin: 20px;
16            padding: 20px;
17            width: 200px;
18        }
19    </style>
20 </head>
21 <body>
22     <div class="container">
23         <div>A. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus id fermentum elit
24         <div>B. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus id fermentum elit
25         <div>C. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus id fermentum elit
26         <div>D. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus id fermentum elit
27         <div>E. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus id fermentum elit
28         <div>F. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus id fermentum elit
29         <div>G. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus id fermentum elit
30         <div>H. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus id fermentum elit
31         <div>I. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus id fermentum elit
32         <div>J. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus id fermentum elit
33         <div>K. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus id fermentum elit
34         <div>L. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus id fermentum elit
35     </div>
```



A. Lorem ipsum dolor
sit amet, consectetur
adipiscing elit. Phasellus
id fermentum elit.
Aenean ornare porta
sagittis.

B. Lorem ipsum dolor
sit amet, consectetur
adipiscing elit. Phasellus
id fermentum elit.
Aenean ornare porta
sagittis.

C. Lorem ipsum dolor
sit amet, consectetur
adipiscing elit. Phasellus
id fermentum elit.
Aenean ornare porta
sagittis.

D. Lorem ipsum dolor
sit amet, consectetur
adipiscing elit. Phasellus
id fermentum elit.
Aenean ornare porta
sagittis.

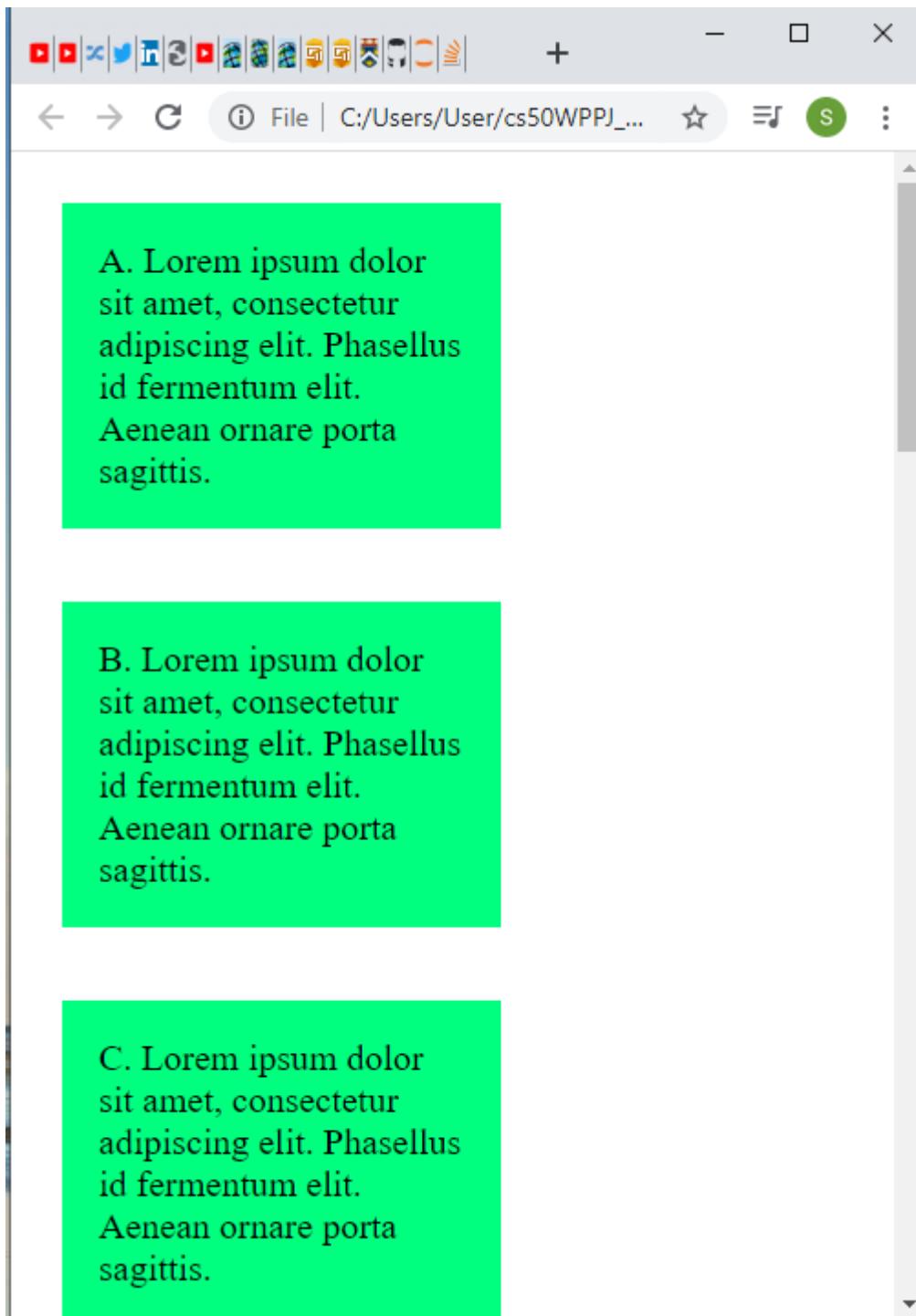
E. Lorem ipsum dolor
sit amet, consectetur
adipiscing elit. Phasellus
id fermentum elit.
Aenean ornare porta
sagittis.

F. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus id fermentum elit. Aenean ornare porta sagittis.

G. Lorem ipsum dolor
sit amet, consectetur
adipiscing elit. Phasellus
id fermentum elit.
Aenean ornare porta
sagittis.

H. Lorem ipsum dolor
sit amet, consectetur
adipiscing elit. Phasellus
id fermentum elit.
Aenean ornare porta
sagittis.

I. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus id fermentum elit. Aenean ornare porta sagittis.



0.0.29 Grid

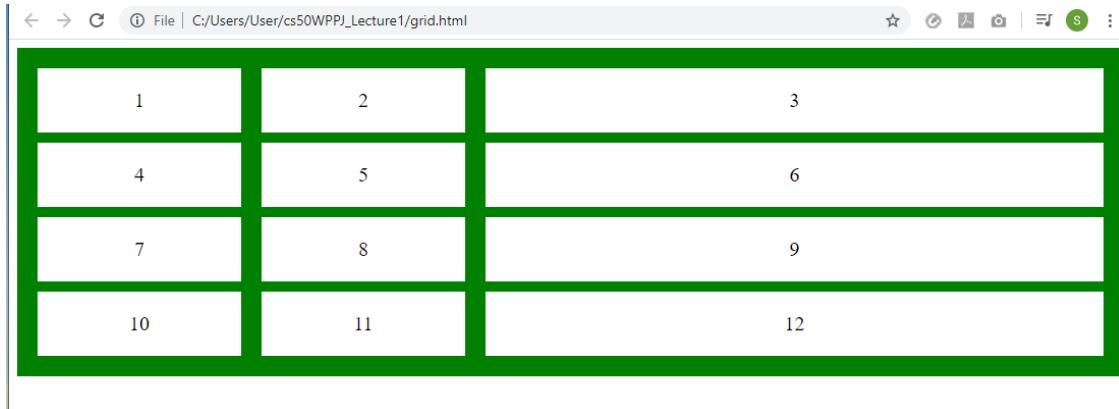
In the above example, the right column changes its size as we change the viewport.

```

<style>
    .grid {
        background-color: green;
        display: grid;
        padding: 20px;
        grid-column-gap: 20px;
        grid-row-gap: 10px;
        grid-template-columns: 200px 200px auto;
    }

    .grid-item {
        background-color: white;
        font-size: 20px;
        padding: 20px;
        text-align: center;
    }
</style>
</head>
<body>
    <div class="grid">
        <div class="grid-item">1</div>
        <div class="grid-item">2</div>
        <div class="grid-item">3</div>
        <div class="grid-item">4</div>
        <div class="grid-item">5</div>
        <div class="grid-item">6</div>
        <div class="grid-item">7</div>
        <div class="grid-item">8</div>
        <div class="grid-item">9</div>
        <div class="grid-item">10</div>
        <div class="grid-item">11</div>
        <div class="grid-item">12</div>
    </div>
</body>

```



Below code will adjust the right two columns.

```
grid-template-columns: 200px auto auto;
```

Below code will adjust all three columns.

```
grid-template-columns: auto auto auto;
```

0.0.30 Bootstrap

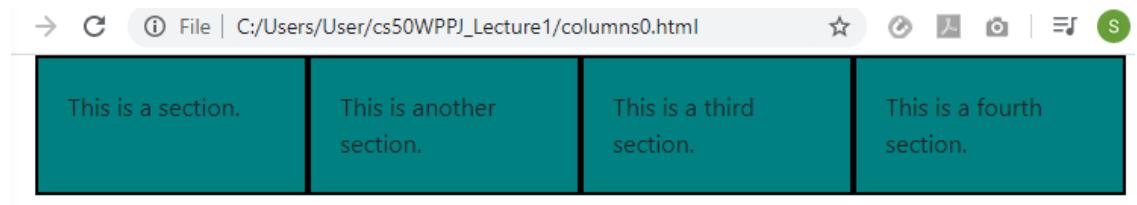
```
<!DOCTYPE html>
<html>
  <head>
    <title>My Web Page!</title>
    <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-Gn5384xqQ1aoWXA+058"
```

Add this line after **title** tag inside **head** tag.

```
<link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-Gn5384xqQ1aoWXA+058"
```

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <title>My Web Page!</title>
5      <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-Gn5384xqQ1aoWXA+058">
6      <style>
7        .row > div {
8          padding: 20px;
9          background-color: teal;
10         border: 2px solid black;
11       }
12     </style>
13   </head>
14   <body>
15     <div class="container">
16       <div class="row">
17         <div class="col-3">
18           This is a section.
19         </div>
20         <div class="col-3">
21           This is another section.
22         </div>
23         <div class="col-3">
24           This is a third section.
25         </div>
26         <div class="col-3">
27           This is a fourth section.
28         </div>
29       </div>
30     </div>
31   </body>
32 </html>
```

Example

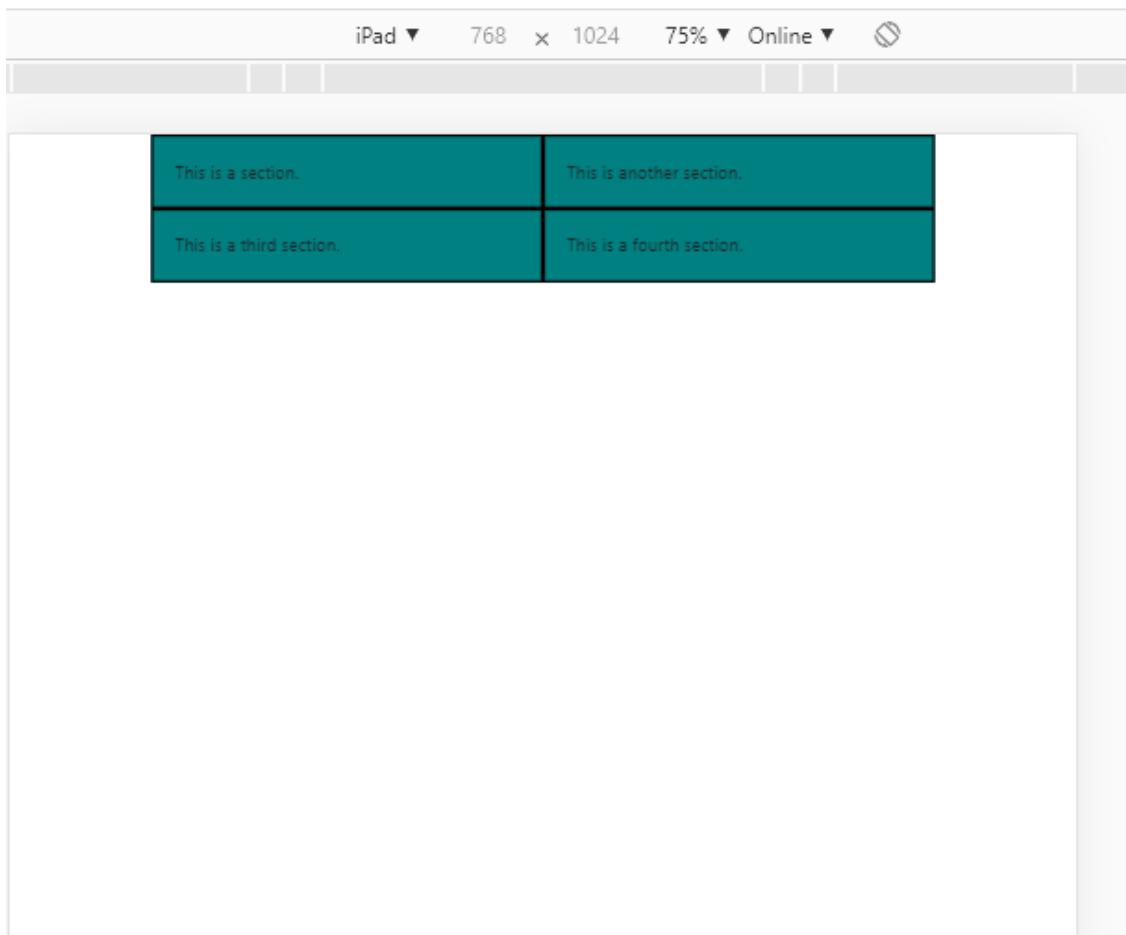


Example of specifying different column layout for different screen sizes

```
C:\> Users > User > cs50WPPJ_Lecture1 > columns1.html > html > head > link
1  <!DOCTYPE html>
2  <html>
3  |   <head>
4  |       <title>My Web Page!</title>
5  |       <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css">
6  |   <style>
7  |       .row > div {
8  |           padding: 20px;
9  |           background-color: #teal;
10 |           border: 2px solid black;
11 |       }
12 |   </style>
13 | </head>
14 | <body>
15 |     <div class="container">
16 |         <div class="row">
17 |             <div class="col-lg-3 col-sm-6">
18 |                 This is a section.
19 |             </div>
20 |             <div class="col-lg-3 col-sm-6">
21 |                 This is another section.
22 |             </div>
23 |             <div class="col-lg-3 col-sm-6">
24 |                 This is a third section.
25 |             </div>
26 |             <div class="col-lg-3 col-sm-6">
27 |                 This is a fourth section.
28 |             </div>
29 |         </div>
30 |     </div>
31 | </body>
32 </html>
```

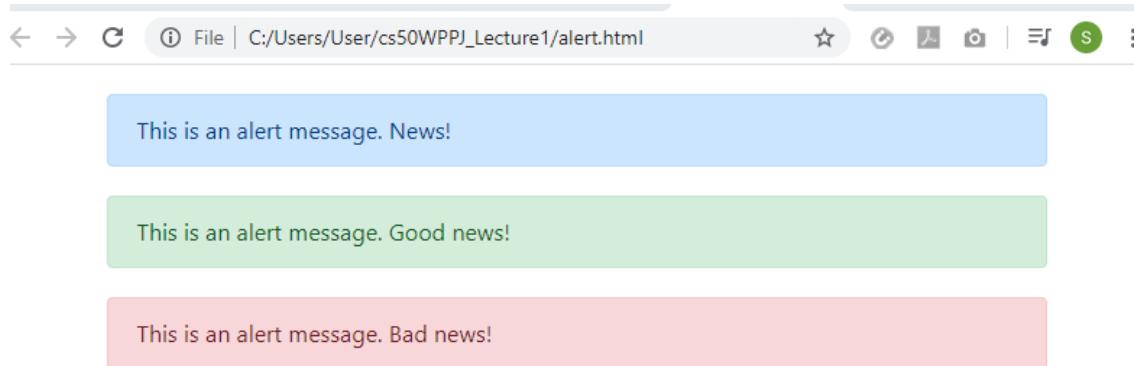
PPJ_Lecture1/columns1.html

This is a section.	This is another section.	This is a third section.	This is a fourth section.
--------------------	--------------------------	--------------------------	---------------------------



```
grid.html  ◊ grid2.html  ◊ grid3.html  ◊ bootstrap.html  ◊ columns0.html  ◊ columns1.html  ◊ alert.html  X  [grid.html]
C: > Users > User > cs50WPPJ_Lecture1 > ◊ alert.html > ⚡ html > ⚡ head > ⚡ link
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <title>My Web Page!</title>
5      <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" integrity="sha384-Gn5BfPvH+OLXqE5ZLNU5tXoDy7dmlZnQD9eT4kS5IwvDzjLW0=" crossorigin="anonymous" />
6      <style>
7        .alert {
8          margin: 20px;
9        }
10     </style>
11   </head>
12   <body>
13     <div class="container">
14       <div class="alert alert-primary" role="alert">
15         This is an alert message. News!
16       </div>
17       <div class="alert alert-success" role="alert">
18         This is an alert message. Good news!
19       </div>
20       <div class="alert alert-danger" role="alert">
21         This is an alert message. Bad news!
22       </div>
23     </div>
24   </body>
25 </html>
```

alert class



0.0.31 SCSS

Define a variable using \$

```
bootstrap.html    columns0.html    columns1.html    alert.html    variables.html    variables.scss <
> Users > User > cs50WPPJ_Lecture1 > variables > variables.scss > ...
1  $color: red;
2
3  ul {
4      font-size: 14px;
5      color: $color;
6  }
7
8  ol {
9      font-size: 18px;
10     color: $color;
11 }
```

A screenshot of a code editor showing a SCSS file named "variables.scss". The file contains SCSS code defining a variable \$color set to red, and then using it in the styles for an ul and ol list items. The code editor interface shows tabs for other files like bootstrap.html and alert.html, and a sidebar on the right.

We need to convert scss file to css using sass

Using **Koala**, compile scss to css.

0.0.32 Using SASS

If we want to use command prompt to compile scss, do the following.

Install ruby from this site. <https://rubyinstaller.org/downloads/>

Now, install **SASS** by executing the following in the command prompt.

gem install sass In the command prompt, navigate to the folder where you're going to write your CSS using Sass syntax.

0.0.33 Tell Sass to “watch” the File

sass –watch variables.scss:variables.css Now, as we change the scss file, automatically it gets compiled to css

Github automatically converts scss to css

0.0.34 Nesting different css selectors within each other

```
C:/> Users / User / cs50WPPJ_Lecture1 / nesting / <nesting.html> ...
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>My Web Page!</title>
5          <link rel="stylesheet" href="nesting.css">
6      </head>
7      <body>
8
9          <div>
10             <p>This is a paragraph inside the div.</p>
11
12             List inside the div:
13             <ul>
14                 <li>item one</li>
15                 <li>item two</li>
16                 <li>item three</li>
17             </ul>
18         </div>
19
20         <p>This is a paragraph outside the div.</p>
21
22         List outside the div:
23         <ul>
24             <li>item one</li>
25             <li>item two</li>
26             <li>item three</li>
27         </ul>
```

```
variables.html    ❀ variables.scss    ❀ nesting.scss X    # nesting.css
> Users > User > cs50WPPJ_Lecture1 > nesting > ❀ nesting.scss > 📁 div
1  div {
2      font-size: 18px;
3
4      p {
5          color: ■ blue;
6      }
7
8      ul {
9          color: □ green;
10     }
11 }
12 }
```

```
# inheritance.css    inheritance.html    inheritance.scss    nesting.html    nesting.scss    # nesting.css X    ...
C: > Users > User > cs50WPPJ_Lecture1 > nesting > # nesting.css > div
1  div {
2      font-size: 18px;
3      div p {
4          color: blue;
5      }
6      div ul {
7          color: green;
8      }
9  /*# sourceMappingURL=nesting.css.map */
```

This is a paragraph inside the div.

List inside the div:

- item one
- item two
- item three

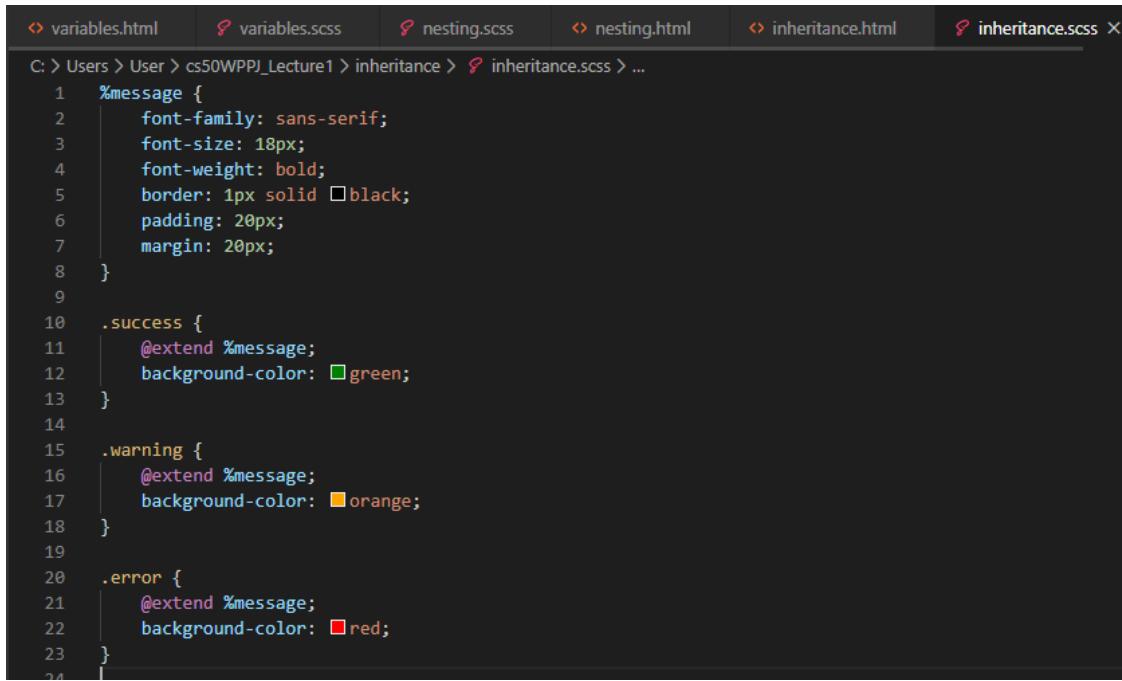
This is a paragraph outside the div.

List outside the div:

- item one
- item two
- item three

0.0.35 Inheritance

```
variables.html    variables.scss    nesting.scss    nesting.html    inheritance.html X    inheritance.scss
> Users > User > cs50WPPJ_Lecture1 > inheritance > inheritance.html > ...
1  !DOCTYPE html
2  <html>
3      <head>
4          <title>My Web Page!</title>
5          <link rel="stylesheet" href="inheritance.css">
6      </head>
7      <body>
8
9          <div class="success">This is a success message.</div>
10
11         <div class="warning">This is a warning message.</div>
12
13         <div class="error">This is an error message.</div>
14
15     </body>
16 </html>
```



```
C: > Users > User > cs50WPPJ_Lecture1 > inheritance > inheritance.scss > ...
1  %message {
2    font-family: sans-serif;
3    font-size: 18px;
4    font-weight: bold;
5    border: 1px solid black;
6    padding: 20px;
7    margin: 20px;
8  }
9
10 .success {
11   @extend %message;
12   background-color: green;
13 }
14
15 .warning {
16   @extend %message;
17   background-color: orange;
18 }
19
20 .error {
21   @extend %message;
22   background-color: red;
23 }
```

Now, compile **inheritance.scss** to css.

```
C:\Users\User\cs50WPPJ_Lecture1\variables>cd ..
C:\Users\User\cs50WPPJ_Lecture1>cd inheritance
C:\Users\User\cs50WPPJ_Lecture1\inheritance>sass --watch inheritance.scss:inheritance.css
>>> Sass is watching for changes. Press Ctrl-C to stop.
      write inheritance.css
      write inheritance.css.map
```

```
# inheritance.css X  ↗ inheritance.scss  ↘ inheritance.html  ↙ nesting.html
C: > Users > User > cs50WPPJ_Lecture1 > inheritance > # inheritance.css > .success
1  .success, .warning, .error {
2    font-family: sans-serif;
3    font-size: 18px;
4    font-weight: bold;
5    border: 1px solid black;
6    padding: 20px;
7    margin: 20px; }
8
9  .success {
10   background-color: green; }
11
12 .warning {
13   background-color: orange; }
14
15 .error {
16   background-color: red; }
17
18 /*# sourceMappingURL=inheritance.css.map */
19
```

This is a success message.

This is a warning message.

This is an error message.

[]: