Git Branch Merge

Merge Branches

We have the emergency fix ready, and so let's merge the master and emergency-fix branches.

First, we need to change to the master branch:

```
=> git checkout master
```

Now we merge the current branch (master) with emergency-fix:

Since the emergency-fix branch came directly from master, and no other changes had been made to master while we were working, Git sees this as a continuation of master. So it can "Fast-forward", just pointing both master and emergency-fix to the same commit.

As master and emergency-fix are essentially the same now, we can delete emergency-fix, as it is no longer needed:

Merge Conflict

Now we can move over to hello-world-images and keep working. Add another image file (img_hello_git.jpg) and change index.html, so it shows it:

We see that index.html has been changed in both branches. Now we are ready to merge hello-world-images into master. But what will happen to the changes we recently made in master?

```
git merge hello-world-images

Auto-merging index.html
CONFLICT (content): Merge conflict in index.html
Automatic merge failed; fix conflicts and then commit the result.
```

The merge failed, as there is conflict between the versions for index.html. Let us check the status:

```
git status

On branch master
You have unmerged paths.
  (fix conflicts and run "git commit")
  (use "git merge --abort" to abort the merge)

Unmerged paths:
  (use "git add <file>..." to mark resolution)
        both modified:   index.html

no changes added to commit (use "git add" and/or "git commit -a")
```

This confirms there is a conflict in index.html, but the image files are ready and stagedto be committed.

So we need to fix that conflict. Open the file in our editor:

Now we can stage index.html and check the status:

```
git add index.html
```

The conflict has been fixed, and we can use commit to conclude the merge:

```
git commit -m "merged with hello-world-images after fixing conflicts"

[master 0db08b9] merged with hello-world-images after fixing conflicts
```

And delete the hello-world-images branch:

```
 git branch -d hello-world-images

Deleted branch hello-world-images (was 2bf7db8).
```

Now you have a better understanding of how branches and merging works. Time to start working with a remote repository!