13. TAGGING VERSIONS

Goals

- To learn how to tag commits for future references

Let's call the current version of the hello program version 1 (v1).

Creating a tag of the first

RUN:

```
git tag v1
```

Now, the current version of the page is referred to as *v1*.

Tags for previous versions

Let's tag the version prior to the current version with the name v1-beta. First of all we will checkout the previous version. Instead of looking up the hash, we are going to use the ^ notation indicating "the parent of v1".

If the `v1^` notation causes troubles, try using `v1~1`, referencing the same version. This notation means "the first version prior to v1".

RUN:

```
git checkout v1^
cat hello.html
```

RUN:

```
$ git checkout v1^
Note: checking out 'v1^'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in
this
```

```
state without impacting any branches by performing another checkout.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -b with the checkout command again.
Example:

  git checkout -b new_branch_name

HEAD is now at 8c32287... Added standard HTML page tags
$ cat hello.html
<html>
  <body>
    <h1>Hello, World!</h1>
  </body>
</html>
```

This is the version with `<html>` and `<body>` tags, but without `<head>`. Let's make it's the v1-beta version.

RUN:

```
git tag v1-beta
```

Check out by the tag name

Now try to checkout between the two tagged versions.

RUN:

```
git checkout v1
git checkout v1-beta
```

RESULT:

```
$ git checkout v1
Previous HEAD position was 8c32287... Added standard HTML page tags
HEAD is now at fa3c141... Added HTML header
$ git checkout v1-beta
Previous HEAD position was fa3c141... Added HTML header
HEAD is now at 8c32287... Added standard HTML page tags
```

Viewing tags with the `tag` command

You can see the available tags using the `git tag` command.

RUN:

```
git tag
```

RESULT:

```
$ git tag
v1
v1-beta
```

Viewing tags in logs

You can also check for tags in the log.

RUN:

```
git hist master --all
```

RESULT:

```
$ git hist master --all
* fa3c141 2011-03-09 | Added HTML header (v1, master) [Marina Pushkova]
* 8c32287 2011-03-09 | Added standard HTML page tags (HEAD, v1-beta)
[Marina Pushkova]
* 43628f7 2011-03-09 | Added h1 tag [Marina Pushkova]
* 911e8c9 2011-03-09 | First Commit [Marina Pushkova]
```

You can see tags (`v1` and `v1-beta`) listed in the log together with the name of the branch (`master`). The `HEAD` shows the commit you checked out (currently `v1-beta`).

‹ 12. Getting older versions

14. Discarding local changes (before staging) ›