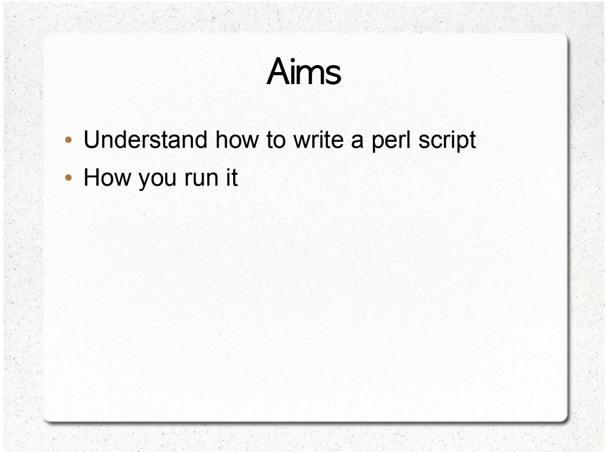
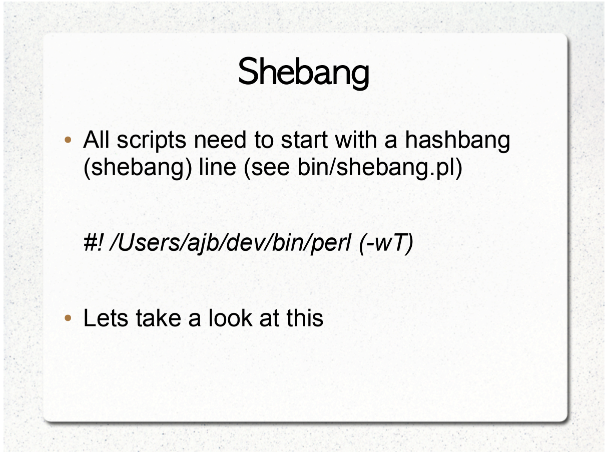
Running a Perl Program



Aims

- Understand how to write a perl script
- How you run it

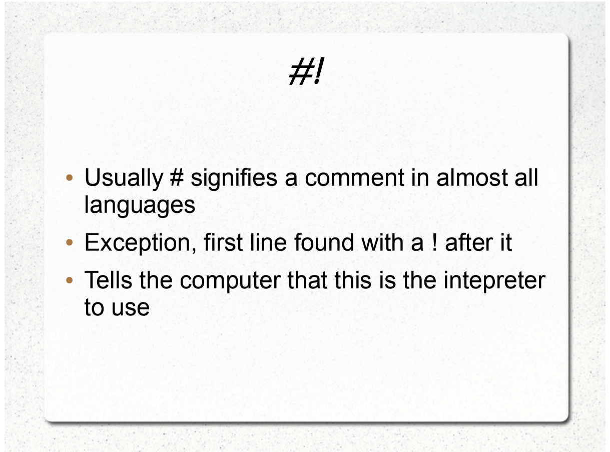


Shebang

- All scripts need to start with a hashbang (shebang) line (see bin/shebang.pl)

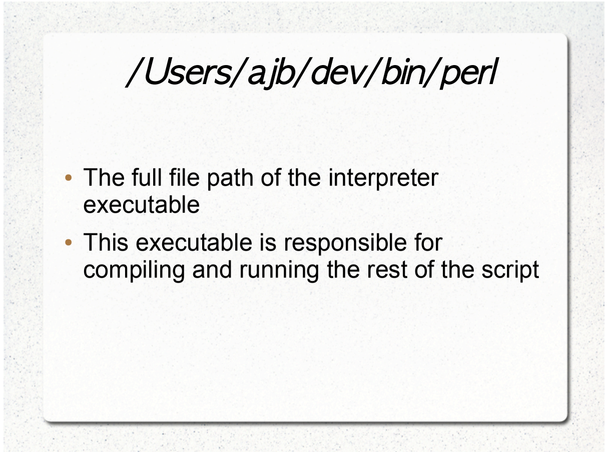
#! /Users/ajb/dev/bin/perl (-wT)

- Lets take a look at this



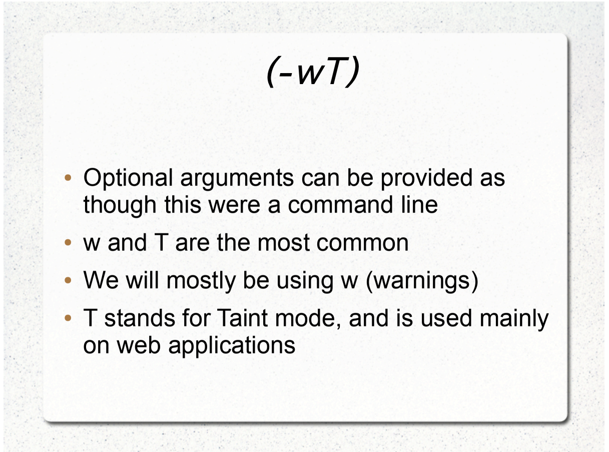
#!

- Usually # signifies a comment in almost all languages
- Exception, first line found with a ! after it
- Tells the computer that this is the intepreter to use



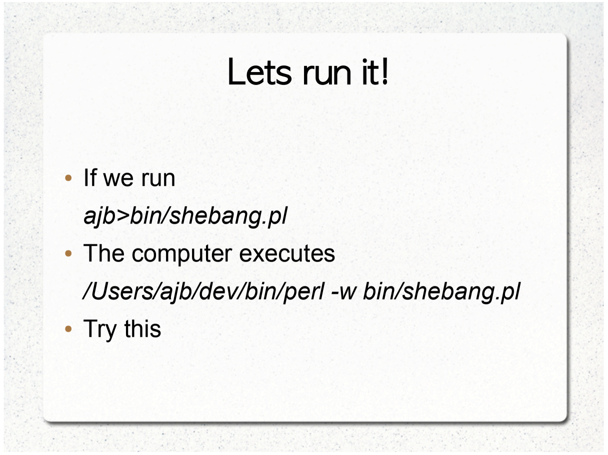
## */Users/ajb/dev/bin/perl*

- The full file path of the interpreter executable
- This executable is responsible for compiling and running the rest of the script



(-wT)

- Optional arguments can be provided as though this were a command line
- w and T are the most common
- We will mostly be using w (warnings)
- T stands for Taint mode, and is used mainly on web applications



Lets run it!

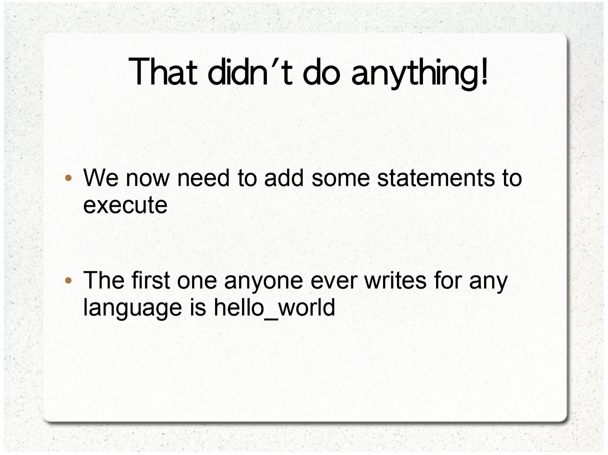
- If we run

```
ajb>bin/shebang.pl
```

- The computer executes

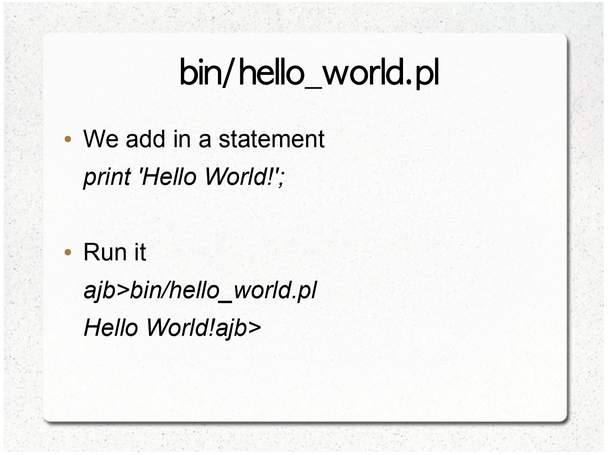
```
/Users/ajb/dev/bin/perl -w bin/shebang.pl
```

- Try this



That didn’t do anything!

- We now need to add some statements to execute
- The first one anyone ever writes for any language is hello_world



bin/hello_world.pl

- We add in a statement

```
print 'Hello World!';
```

- Run it

```
ajb>bin/hello_world.pl
Hello World!ajb>
```

Lets look at the statement.

print

- print is a 'function' or 'keyword'. It tells perl to print the next argument

'Hello World!'

- A string argument, that we want printed

;

- commands need a ; to signify that the command has ended – leaving it out will break your code

comments

- Look in the file

bin/hello_world.pl

- *You'll see*

# this is a comment to explain print (do this function on) 'Hello World!' (this string) ; (end of statement)

- Anything after a # until the end of the line is a comment, and is ignored

Failing scripts

- Perl tries to be helpful when scripts fail

ajb>bin/fail_world.pl

Illegal division by zero at bin/fail_world.pl line 4

- This is helpful. We know what the error is, and where it is in the script.
- Sometimes some investigation further is needed, but it helps.

External Code

- You can pull in extra code (modules) to provide extra functions

use Modern::Perl;

say 'Welcome, to a whole new world.';

- The *use* keyword tells the interpreter to go and locate this code, and compile and run it, and gives us *say*

External Code

ajb>bin/brave_new_world.pl

Welcome, to a whole new world.

ajb>

- Try removing the use line, save and retry
- BIG ERROR message – say function doesn't exist

Summary

- You must have the shebang line:

#! /Users/ajb/dev/bin/perl (-wT)

- Statements follow

<function> <parameters>;

- Use external code

use Modern::Perl;

- Failures fail, and try to be helpful

Any questions so far?