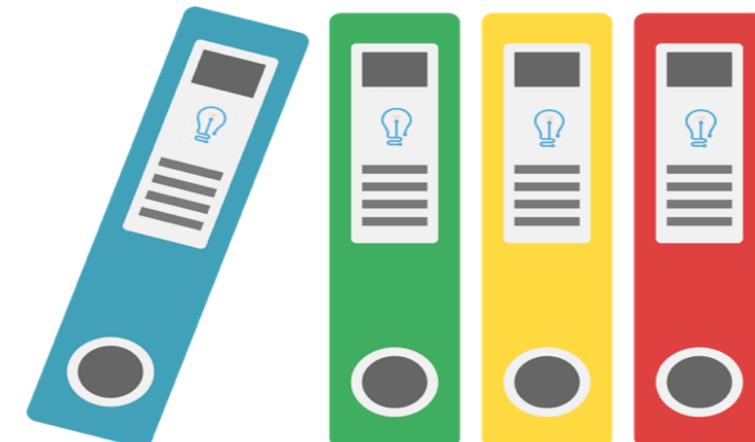


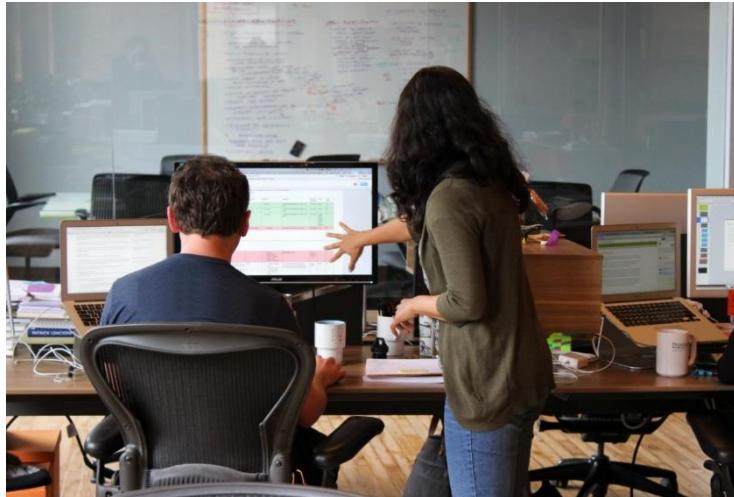
Documenting Libraries



JavaDoc



Different Audiences for Documentation



End User



Library User



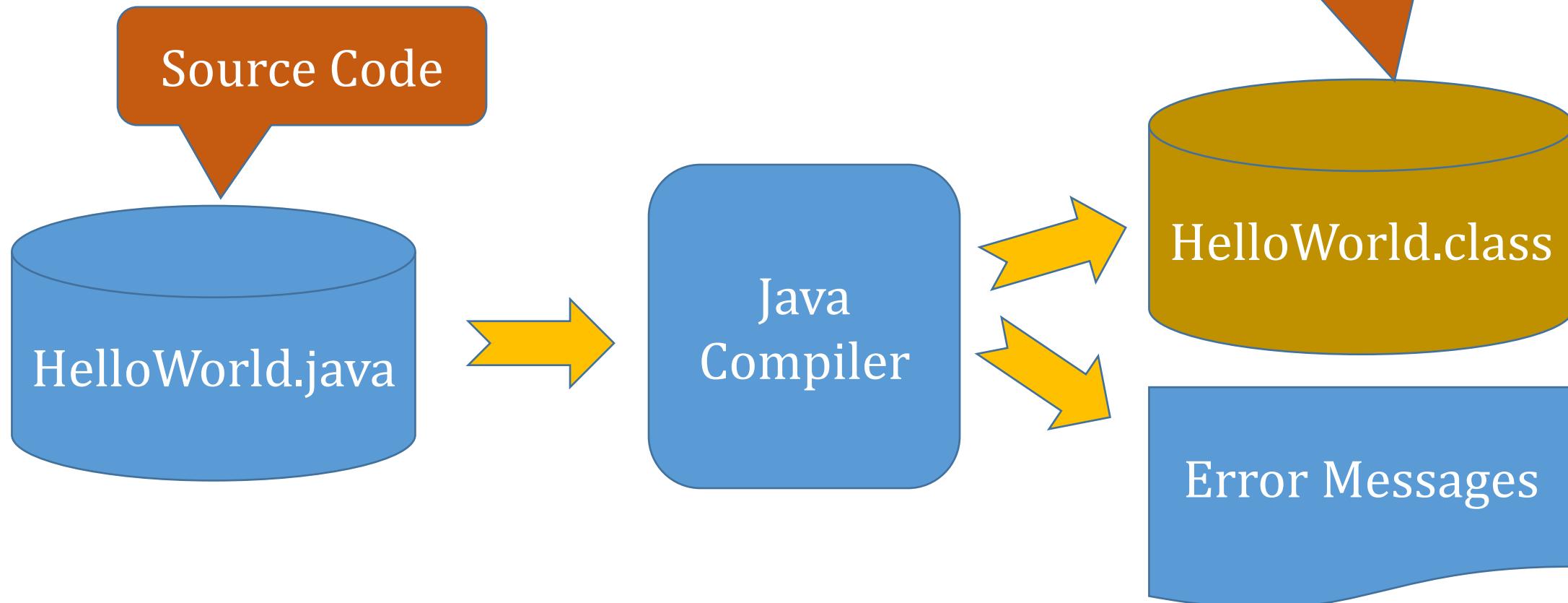
Programmer



Javadoc...

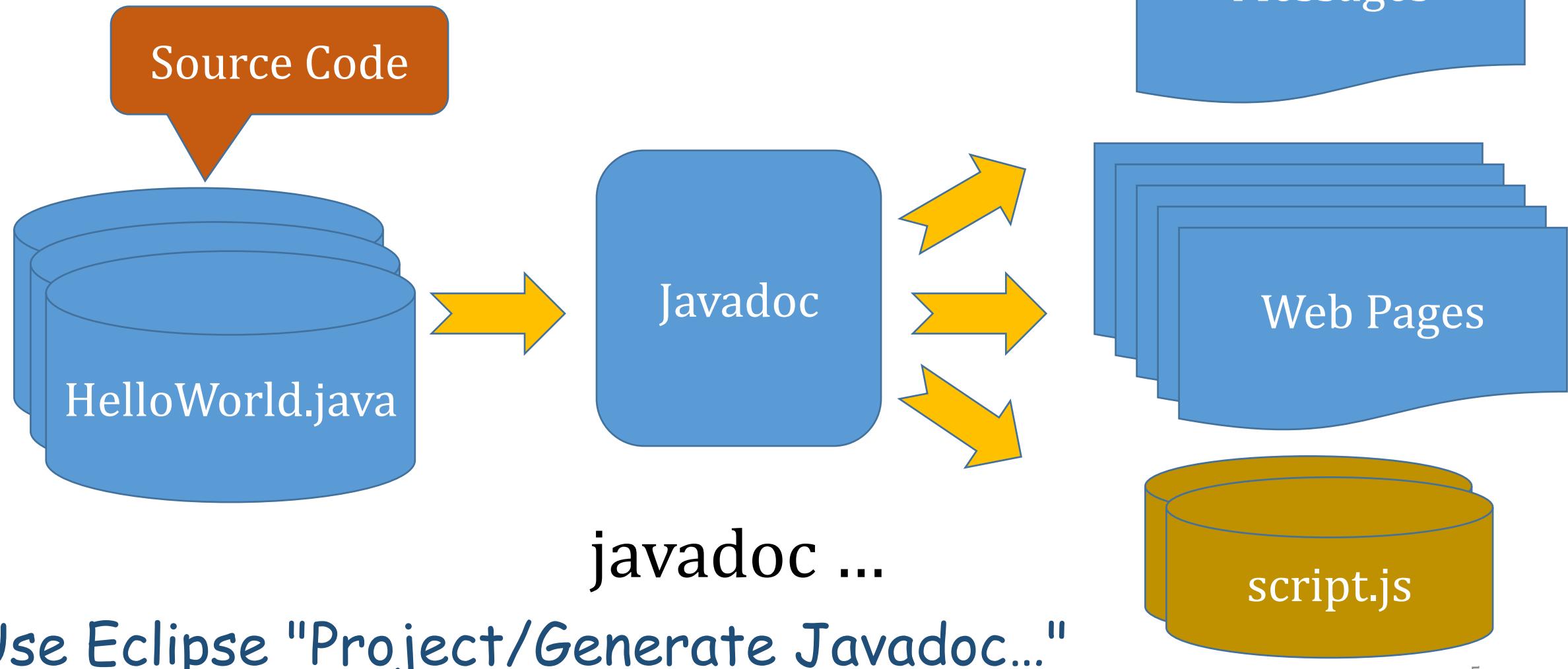
- Developed to document the Java libraries
- Concept... embed the documentation in the code
 - Handy to update when code changes
 - Unique, recognizable format of comment
- Creates web pages in .../doc subdirectory
- To display those web pages:
 - Eclipse can display, or
 - Web browsers can display local html files
 - Or, put the files in a place accessible by a server

Compiling Java Code



`javac HelloWorld.java`

Running Javadoc



Javadoc Comments

```
/**  
 * Text goes here  
 *  
 * And can be continued as long as required  
 *  
 * but ends with...  
 */
```

Open comment
Must be on it's own line

As much text as you want
This text will be used by javadoc

End comment
on it's own line

Use Eclipse "Source/Generate Element Comment"

Where does javadoc look for comments?

1. Above the "class" statement to define the entire class
2. Above each accessible constructor and method
 - Javadoc parameters specify what is "accessible"
 - For instance, you may not want "private" methods documented
3. Above each enumeration constant
 - Enumerations yet to come
4. Non-standard javadoc templates may use other comments
 - For instance, javadoc comments above field declarations

javadoc comment contents

- Comments starts with a "description" in html
 - First sentence is the "summary"
 - For the most part, raw text is good enough
 - White space will be ignored (other than an entire blank line)
 - Can contain "inline tags"
 - Description ends with end of comment or a line that starts with "@"
- After the description, add "tags"...

javadoc tags

- **@author *yourName***
 - Required for classes and interfaces
- **@param *name* *description***
 - Required for each method/constructor parameter
- **@return *description***
 - Required for each non-void method
- **@throws *exception* *description***
 - we'll talk later

Referencing related things

- `{@link package.class#member}`
- Generates a hyperlink to the specified package, class, member.
- package not required if it's the current package
- class not required if it's the current class
- "member" is a method, constructor, or field name
 - include the parenthesis and argument names

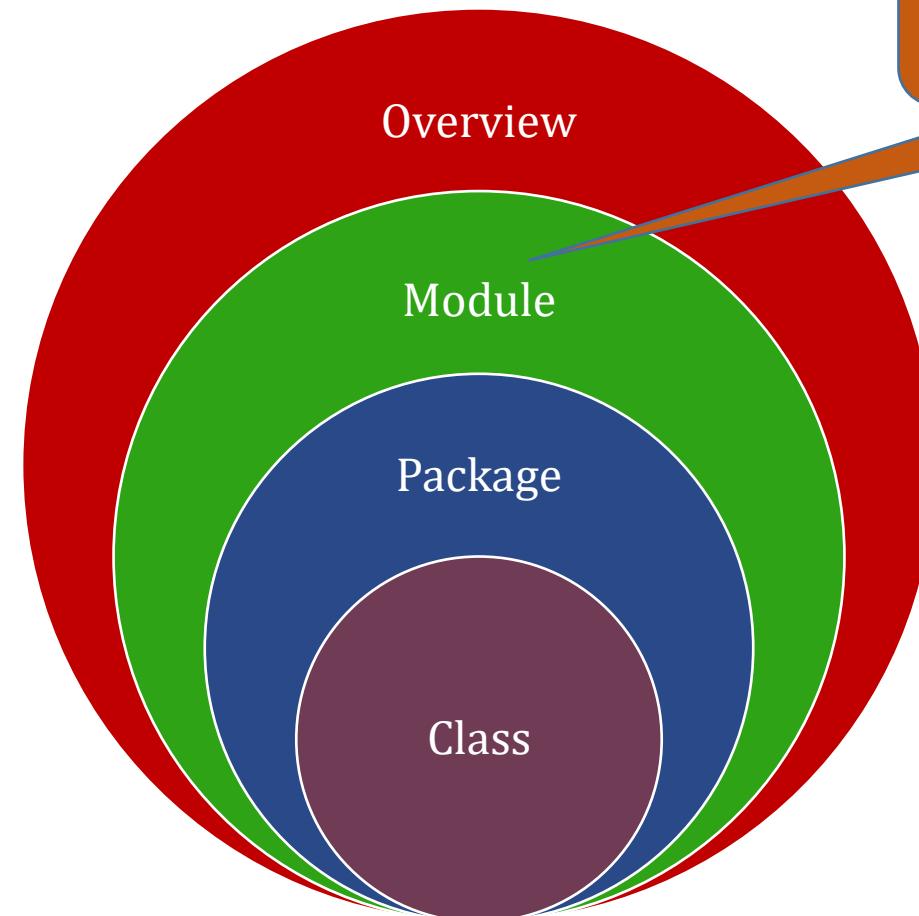
```
/**
```

```
* Like the {@link #add(add)} method, ...  
*/
```

Package overview Documentation

- Optionally, create a "package-info.java" file in the package
- The file contains a package statement preceded by a javadoc comment
- The javadoc comment is the package level documentation

Javadoc HTML Hierarchy



We won't use module
~Eclipse project

Why Javadoc?

- Standardized documentation for you to use
 - All java library documentation is created by javadoc!
- Standardize comments in source code
 - Know where to look for method and parameter documentation
- Documentation for programmers to use your java library code
 - Not terribly useful in an academic setting
 - Much more useful in "industry"
- Not useful for "end-user" documentation!