

# Intro to Computer Programming: week 11

This week's homework:

Synopsis:

- Write a Perl cgi program that produces an HTML page by downloading a Javascript file with functions in the file.
- Read HTML pages on HTML layout tutorial at w3schools
- Read perl CGI excerpts on adding Javascript files to your HTML pages

Details:

1. Write a Perl cgi program that produces an HTML page by downloading a Javascript (JS) file with functions in the file.
  - a. Our previous assignment was to embed the JS in the HTML directly. This assignment is to use an external JS file instead of the code being embedded.
  - b. We will do this by using Perl CGI to create an HTML page with two 'div' sections with nothing in them. The HTML page produced will also have in the HTML header section tags to use an external JS file that you will also produce and place on your web site.
  - c. Then once the HTML page loads it will execute a function from your JS file to add content to one of the 'div' sections.
  - d. Once assignment is done email me.
  - e. Steps to produce HTML page:
    - i. Copy one of the existing CGI programs and name it `first_js_file.cgi`.
    - ii. Edit it (remove existing content) so it produces just a plain web page with a new title and an HTML H1 line at the top with same content as the title. (The `process_form.cgi` is a good choice. Note you might have called this program something different.)
    - iii. Upload this file to your cgi-bin directory.
    - iv. Add a link to your home page to this program. Call the link 'My first JS file'.
    - v. Create a text file for your JS functions called 'first.js'.
    - vi. Upload this file to the home directory of your web site. NOT in your cgi-bin directory
    - vii. Add to the Perl program `first_js_file.cgi` the code to download 'first.js' with your produced web page.
      - This is done in the Perl CGI function call 'start\_html'

- viii. Add to Perl program `first_js_file.cgi` the code to produce two div sections.
    - Add an id tag to each div. The first one with `id=top` and the second one with `id=bottom`
  - ix. Add a function, called 'top\_div', to the JS file `first.js` that adds any desired HTML to the div section with `id=top`
  - x. Add to Perl program `first_js_file.cgi` the code to run this JS function when the onload event is triggered.
    - This is done in the Perl CGI function call 'start\_html'
2. Read HTML pages on HTML layout tutorial at w3schools
    - a. Read: [http://www.w3schools.com/html/html\\_layout.asp](http://www.w3schools.com/html/html_layout.asp)
      - i. The first example is layout by use of HTML tables. This is considered 'old-school' as it was the first method used when the internet was young.
      - ii. The second example is layout by use of HTML 'div' tags that create sections in an HTML page.
    - b. Read: [http://www.w3schools.com/tags/tag\\_div.asp](http://www.w3schools.com/tags/tag_div.asp) for info on the DIV tag.
      - i. Make sure you try the 'Try It' example to ensure you understand how the div tag works.
      - ii. Also read the page on Events so you understand how they affect the div tag:
        - [http://www.w3schools.com/tags/ref\\_eventattributes.asp](http://www.w3schools.com/tags/ref_eventattributes.asp)
    - c. Read: [http://www.w3schools.com/js/js\\_functions.asp](http://www.w3schools.com/js/js_functions.asp) for info on JS functions
  3. Excerpts from Perl do on the CGI module needed for this assignment
    - a. Found at: <http://perldoc.perl.org/CGI.html#CREATING-THE-HTML-DOCUMENT-HEADER>

## CREATING THE HTML DOCUMENT HEADER

```
1. print CGI::start_html(-title=>'My first JS file',
2.
3.           -script=>{ -type=>'JAVASCRIPT', -src=>'/first.js' },
4.
5.           -onload=>'top_div()'
6. );
```

The `start_html()` routine creates the top of the page, along with a lot of optional information that controls the page's appearance and behavior. This method returns a canned HTML header and the opening `<body>` tag. All parameters are optional.

## CREATING THE HTML div Section

```
print CGI::div( { -id=>'top' } );
```

