

Part 1

Copy these four questions into a new Word document and answer them in **long-form**.

1.1 Describe in your own words how the web works! In as much detail as you can, describe **all** the sequences of events that take place from the time a user presses Enter on the keyboard after typing in www.rpi.edu into the address bar to when the webpage is finished rendering in the browser. Specifically, tell me in great detail the protocols in action. (10 points)

DNS lookup

- The first thing that happens when the user presses enter is the users machine looks at the hosts file (located in C:/windows/system32/drivers/etc/hosts for windows machines and etc/hosts for unix machines) to see if the domain is already there. If it is, then you are sent to whatever ip address is specified in the hosts file for that domain name and the DNS lookup is done. Otherwise, we move on to the next step.
- If the user is on an enterprise network, then your machine checks the enterprises' DNS cache server which functions similarly to the hosts file mentioned earlier. If something is found, then we go to the ip address and the DNS lookup is done.
- The next thing we do is move to the root or top level domain server with the query for "www.rpi.edu". The server responds with the address of a more specific server, the nameserver to be exact, which contains information about all of the websites ending in .edu in our case.
- The nameserver does not return the address of our website but another server, this time the actual server for www.rpi.edu. Now, our DNS lookup is complete and we have the ip address that allows us to perform a HTTP transaction.

HTTP transaction

- We will be sending a request header with verb "GET" to get the HTML for the page that we can then load onto our browser.
- We first establish a TCP connection to the ip address that we received in our DNS lookup
 - We do this by first sending and receiving packets.
 - The first packet we send is a SYN packet to the ip address
 - We then wait for a response of a SYN-ACK packet from the ip address
 - And then after we receive the SYN-ACK packet we send an ACK packet.
 - This way we know that we are properly sending and receiving data to and from the ip address.
- After we establish the connection, we send a request header to the server.
 - In this case we are just visiting www.rpi.edu so we will be using the GET verb.

- GET lets us retrieve resources from the specified domain
- There are lots of additional components, but the ones absolutely required are:
 - The verb (GET)
 - The path (/)
 - The host (rpi.edu)
- There are many other things that are added but those 3 let us say that we want to get the page at the directory / from rpi.edu.
- We then receive the response header
 - The initial response header basically just tells us that it works (it gives us status code 200 which means good) and the HTML of the site.
- After the initial request and response headers more are sent and received which give us more components of the site
 - For instance, if we need to use CSS or javascript from external files we need to get them somehow

1.2 What is the difference between a property and a method in JavaScript? (3 points)

Objects in javascript are essentially key-value pairs stored in a hash, and the difference between a property and method for an object is a property is just a value stored in a hash key while methods are functions stored in a hash key.

1.3 Explain how your browser chooses which CSS rule to apply to a tag in the case where there are multiple rules that could apply. (3 points)

Our browsers use a term called Specificity, which determines the order that CSS rules apply and when they override other rules. As per the MDN docs,

“Specificity is a weight that is applied to a given CSS declaration, determined by the number of each selector type in the matching selector. When multiple declarations have equal specificity, the last declaration found in the CSS is applied to the element. Specificity only applies when the same element is targeted by multiple declarations. As per CSS rules, directly targeted elements will always take precedence over rules which an element inherits from its ancestor.”

This essentially means that the more specific our selectors are when creating rules for them, the higher priority they have when the rules are being applied - In the example below, the text will always be green because of the most specific rule.

```
<div id="test">
  <span>this is some text</span>
</div>
```

```
div#test span { color:green; }
div span {color:red; }
span { color:blue; }
```

1.4 State **four** total advantages of “separation of concerns,” for any permutations of that term we discussed in class. (4 points)

1. Because the javascript/CSS is in a different file compared to the HTML, team members can work on the same project but on different parts at the same time without stepping on each other's toes by working in separate files. This also works great with VCS like github, where there are separate branches that team members can work on and they can just merge them to main and then have the other branches then update their branches from main to see the changes from the merged branch without having to change what they're working on.
2. Again, because the languages have been segregated, the project overall becomes more maintainable because there is no single 2000 line HTML document with all of the CSS and javascript all combined in the head.
3. Because of the increased maintainability, the system then is more stable.
4. Lastly, separation of concerns reduces the overall complexity of the project and improves the ability for someone to understand how it works by breaking the project down into many smaller and digestible bits and pieces.

Extra Credit (+5 points)

1. In what year did Prof. Plotka graduate from RPI? What was his major?
1987, B.S. in Computer Science