Q1. In https, what is the initial sequence of steps, to establish a secure communication channel?
A1. Client contacts server for its public key; client generates a session key, encrypts it with the server's public key, sends that to the server. [Now both client and server have the session key, which they use for symmetrically encrypted comm.]

Q2. Name, and explain in a sentence each, two JavaScript features/aspects that are absent in other programming languages such as Python, Java, C++ etc.
A2. Prototypical inheritance, immediately invoked function execution (IIFE), a single 'Number' class, an 'arguments' array accessible inside any regular (not 'arrow') function, etc.

Q3. Mention and briefly discuss (with a sentence each) at least two characteristics that make JSON an ideal format for data specification.
A3. K:V pairs make it possible to turn column names (in tabular data) to keys; arbitrary nesting of arrays and objects makes it possible to describe data of arbitrary/unlimited nesting.; plaintext, uncluttered format makes it human readable; the simple format makes it ideal for any data generator to output its data as JSON (eg. REST API calls, scientific instruments, DB lookups...).

Q4. What is the main reason why DNS exists? Explain in a few lines, with an example.
A4. Because humans better remember website names (eg. google.com) as opposed to IP addresses (eg. 209.85.128.0).

Q5. What is the big advantage in representing a web page's elements, in a structured tree form, ie as a DOM? Be specific.
A5. We can programmatically walk the DOM tree and alter style/content anywhere in the document [can modify any element's style or value selectively by accessing it using its path or ID or type], or create elements programmatically (eg as done by frontend frameworks) to insert into the DOM, can show/hide elements selectively... The key is 'programmatic access'.

Q6. What does XHR permit? In other words, what makes it useful? Explain with an example.
A6. XHR (XmlHttpRequest) lets us request data from a server, from within a web page, using GET or POST - the response can be used to dynamically populate views in the page, ie. it lets us selectively update parts of a page, without having to re-render the entire page.

Q7. What is the most 'popular' server response that even the lay (non technical) public would know about, and when does the server generate it?
A7. '404 Page Not Found' - when the client requests an invalid URL.

Q8. What is the predecessor of fetch(), and how does fetch() improve on it?
A8. fetch()'s predecessor is XHR, ie XmlHttpRequest. XHR uses callbacks, which become extremely hard to follow/debug when multiple calls need to be made; fetch() uses Promises instead, making for much cleaner syntax: fetch().then().

Q9. What purpose/function does a 'certificate authority' ('CA') serve?
A9. The role of the Certificate Authority (CA) is to guarantee that the server granted the unique certificate is, in fact, who it claims to be - so it makes browsing safer.

Q10. What are some mechanisms using which, a webserver is able to serve more than just HTML documents? Name, and briefly discuss,at least two.
A10. cgi-bin server scripts, ASP, JSP, database lookups, REST API calls...

Q11. What are some similarities, and differences, between HTML and XML?
A11. Similarities: both are based on key-value pairing; both can describe arbitrarily nested (deep) hierarchies. Diffs: XML uses tags, JSON uses k:v syntax; XML needs matching tags, JSON does not; XML makes a distinction between attribute values (attribute themselves are part of an element) and element values, JSON does not.

Q12. What advantages do cloud servers (eg. Azure) provide, over 'traditional' ("dot com") servers?
A12. 'Managed services' - the cloud is maintained by the cloud provider, who can guarantee uptime, provide 'unlimited' (elastically-scaled) computing and storage, can keep the cloud infrastructure up-to-date...

Q13. What are the two mechanisms by which a client makes a request to a server, when is each used?
A13. GET, for simple data sent to the server (eg a search term); POST, for more complex data, eg. a form's structured contents.

Q14. When uploading a Python app to a cloud server, why do we also include a 'requirements.txt' file?
A14. It serves as a 'packing slip' - the cloud server would use it to INSTALL modules listed in the requirements file so that the app that needs to run on the cloud can do so properly. In brief, it is to ensure that an app's dependencies are satisfied, on the cloud.

Q15. What is HTTP built on? Explain using a few sentences, and a diagram.
A15. Built on underlying mechanisms - SSL, TCP/IP.

Q16. What is the following form of JS code called?

```
let t = (function () {
  return 0.5*(Math.random()+Math.random());
}());
```

A16. IIFE. One use - to create a namespace (eg JSON) and populate it with functions. Other uses - to create a module with publicly exported functions, to be used to dynamically populate a web page with elements, etc.

Q17. In the context of a web app, what is an API? Explain, in a couple of sentences.
A17. An API is a collection of web service calls that a client can make, using a simple URL with extra query parameters attached at the end; the response can be JSON, XML, plaintext, HTML, blob...

Q18. What is JavaScript's main purpose? Please be specific.
A18. To dynamically create, extend, modify or populate (ie set values) a web page (ie programmatic DOM manipulation - both structure and content).

Q19. Communication from a client to a web server, and a web server to a client, are structurally similar - in what sense?
A19. Both have three parts - a summary (request line, start line), header, body.

Q20. How does WSDL get used?
A20. WSDL is often used in combination with SOAP and an XML Schema to provide Web services over the Internet. A client program connecting to a Web service can read the WSDL file to determine what operations are available on the server. The client sends a WSDL-formatted request, receives a WSDL-formatted response from the server.

Q21. What is a big difference between HTTP/1 and HTTP/1.1?
A21. HTTP 1.1 enables persistent connections, that way there can be more than one request/response on the same HTTP connection.

Q22. What is a 'blob' response type from a server, useful for?
A22. A blob a file-like object of immutable, raw data, which means it can contain music, video, images, array data etc for example; the client can parse the data to extract content it wants.

Q23. In what sense is a callback function, not like a regular function? Be specific.
A23. A callback function is not called in a specific sequence like in a regular piece of code - instead, it's event-driven, ie called in response to a user action (eg via a UI), or some other event (eg an incoming file).

Q24. What's one use for base64 encoding?
A24. It can be used to encode an image - as a result it can be embedded in a web page the server sends,

as part of the page data - this eliminates the need for a separate connection just for the image.

Q25. What are two uses (functions) of a proxy server?
A25. Caching, load balancing (and masking user presence on the web, etc.)

Q26. In structuring a document (eg a web page), why separate form and function (ie. style and content)?
A26. Because one can be modified without affecting the other.

Q27. In what sense can a modern web app be seen as a generalization of classic software development?
A27. The app makes distributed function calls, where a call is made to a server (as a REST API call, for ex) - rather than the app running entirely on the client, it runs in a truly distributed fashion, in servers accessible through the web.

Q28. In a language such as Python or C++ etc, the following set of function calls will execute, in sequence - but not necessarily, in JS:
x = a(...)
y = b(...)
z = c(x,y)
Why not? How would we fix that (ie ensure it)?
A28. Because the calls could involve XHR or fetch(), which are not guaranteed to terminate in a given amount of time (or terminate at all, on account of error) - so functions are all invoked in sequence but are not expected to finish in sequence. To fix it, we could make the functions be chained using .then(), ie. a().then(b()).then(c())

Q29. How are we able to have alternatives to HTML, CSS and JS, running in the browser?
A29. By creating a parser for them, written in JavaScript - the parser can then parse the alternative syntax [eg written in Processing, LaTeX, less, RSX...], and emit standard HTML/CSS/JS for the browser to use "as usual".

Q30. What are two new things you've learned from the course thus far? Explain each using a sentence or two.
A30. :) :)