

Purpose of session: user-friendly session to help us understand what the university is looking for. This is student-led so is our understanding and not official.

Tutor webinar stressed 3 major areas:

(pure) HTML incl semantic tags,

CSS property: value,

Vanilla JS with no frameworks, no compactifying or minifying, with lots of comments

Rubric (read these notes in conjunction with annotated rubric from Slack):

Site1 - working URL hosted on the static web page system. Easy points. No dead pages. Check that the site is being displayed within their engine

Site2 - 5 page tree structure. Make the marker's life easy. Easy to navigate the zip file (must match site). Everything can be linked from home page, or you can have subpages. Content on each page must be different.

Site3 - Accessibility. Both design and report. Alt image attribute, contrast checker, dark mode, font scaler for font, tabbable page for non-mouse user, sticky navbar, other UX concerns are site load times, ease of using site

Site4 - there was a question in the tutorial about creating original content. Not expected to create own images (ex. unsplash.com) or videos. Source reference required. Could present media such as videos in a way that fits with the feel of the site. For text, would suggest writing our own but not a massive amount as we're not being judged on writing ability, but just having written something original that's appropriate: e.g. a small blurb (snippet) rather than a full article which we write, then maybe a link to an external website with a full article.

HTML1 - use validators (W3C Schools has one)

HTML2 - these are often a personal choice. Use HTML5 semantic tags where possible. Only use a div tag where there's no semantic tag from HTML5 available.

HTML3 - Comments are heavily stressed in the tutorial. What do sections represent? (Not just where sections start and end) Descriptions of what's inside the tags and maybe the location of the articles, for example. Imagine Future You coming back to your code and working out why you made your choices. Justify why you made your choices.

CSS1 - Colors.io is an example of a site you can use for generating colour schemes. Suggest only a few colours, maybe black and white and a couple of accent colours. Don't use same layout on all pages, but not every page needs to be different

CSS2 - "Make sure your layouts are not boring" - Dr Ali direct quote. Header and footer should be consistent. Mix up columns, e.g. 2 columns on one page, 4 columns on another - variation in layouts is encouraged. Promotional banners maybe (referencing the source).

CSS3 - Responsive, i.e. desktop and mobile versions. You can test responsiveness using developer tools in the browser. You can state viewport size you tested with in the report.

CSS4 - see HTML comment comments above. Purpose of code.

JS1 - 2 to 3 DOM manipulations (custom made without libraries).

JS2 - you need to make JS a critical part of the site, e.g. image carousels, checking forms, animations.

JS3 - Advanced use can be use of a library, e.g. using a library beyond readme or tutorial. Check for errors and that the library doesn't have any unwanted side effects

JS4 - comments: see previous comments about comments

Report1 - 2 wireframes (simple sketches with no details, margins, colours; just include HTML tags) for each page (desktop and mobile versions). Site map can be a chart (flow chart or hierarchy chart). Wireframes folder.

Report2 - Mockups are like screenshots of the finished page. 2 per page. Mockups folder. Colours, looks like a finished product.

Report3 - See report spec linked in Slack. Evidence of how you made your decisions and tackled the project. Plain HTML, can be without styling, but sections need to be clearly separated. No word limit.

Question: How many points do I need to pass? 30/36 would be 83%. $83\% \times 0.7$ would be 58% so that is a pass. However you also need to pass the midterm as well. Advice is to check the prog regs. <https://london.ac.uk/sites/default/files/regulations/progreps-general-2020-2021.pdf> Go as high as you can, it can be hard to work out the exact number you need to pass. It's also different for PBA student vs non-PBA. Get the easy points first. Mockups, wireframes, static URL.

Question: Handlebars is not a library, it's a templating system - is that correct? Yes, that's correct. No points for Handlebars. For libraries, they are looking for external .js files.

Question: Can someone who can barely write HTML make it for this project? Someone is just starting to learn HTML / CSS now (3 weeks to deadline). We would suggest you maximise your

time with YouTube tutorials (code-along). We will link suggested videos - <https://www.youtube.com/watch?v=mU6anWqZJcc&t=6605s> (FreeCodeCamp). HTML might be a couple of hours, CSS maybe Flexbox Froggie / CSS Diner. Mozilla MDN as well for HTML/CSS/JS.

Question: How do you start off with coding a JS feature? One student said: Start off with your own code until you hit a wall, then look online for solutions to the problems you're having and keep searching. Look for other code that does the same and make small changes, for example scroll up before navigation bar shows, change dimensions, variable names. Enhance or improve the code.

Website colour theme generator: Coolors.co

Color Contrast Checker: <https://contrastchecker.com/>
<https://contrast-ratio.com/>