

# CS2204 Fundamentals of Internet Applications Development

## Lecture 12 Course Review

*Computer Science, City University of Hong Kong*  
*Semester B 2024-25*

# CW2 Reminder (Due on Apr 18th)


CW2 takes **13%** of the course assessment and will be due in one day on **April 18th, 23:59 PM**

Please **start as soon as possible** as they are 6 tasks in this assignment!

If you have any questions about assignment requirement, please **contact the student helpers for clarification.**

If your question is about how to handle a specific task or debugging, the helpers are not supposed to directly help you with that, but they may provide some hints if applicable.

# LOQ Reminder



Office of the Provost

About TLQ ▾ Staff Students FAQ ▾ Contact Us

Login


## Information for Students

To help the University continuously improve teaching and learning and inform personnel decisions, your response truly matters. Completion of the questionnaires is completely on voluntary basis. Please provide an accurate overview of the quality of teaching and learning experience.


Before you start completing the TLQ, you may find the following materials useful.

1. [User guide for completing TLQ](#)
2. [Questionnaire sample](#)
  - The name of the teacher for the section under evaluation will be displayed on the questionnaire.
  - If no teacher name is shown, that means the section has chosen the "TLQ for Courses". The evaluation will be on the section as a whole instead of individual teachers.
  - "Teacher" broadly refers to the one who leads the section. "Teacher" can be a lecturer, tutor, speaker, trainer, mentor, as similar.

To protect your personal data privacy and to preserve data integrity, no personal identifiers will be revealed in the TLQ System, and only eligible users are given the right to access to the reports for academic quality improvement and possibly personnel decisions. You can complete and submit each questionnaire ONCE only. Submitted responses CANNOT be edited or deleted from the TLQ System. Therefore, please complete the questionnaires with great care.



User guide for completing TLQ



Complete TLQ now!

<https://onlinesurvey.cityu.edu.hk/loq/>

# Agenda

Mid-term question review (continued from Lec 11)

Course review

Programing practice

# Agenda

Mid-term question review (continued from Lec 11)

Course review

Programing practice

## Q24. CSS animation (47%)

**What property is used to create animations in CSS3?**

## Q25. CSS animation (53%)

**Which of the following statement about CSS animation is WRONG?**

## Q26. CSS selector (41%)

**Which of the following is an example of a CSS selector that selects an element based on its position relative to other elements?**



## Q27. CSS styling (51%)

**Which of the following is a correct CSS syntax to make all <p> elements bold?**

## Q28. CSS styling (58%)

**What is color of the text X, Y, and Z?**

## Q29. CSS pseudo class (47%)

**Which of the following is true regarding the pseudo class in CSS?**

## Q30. CSS transition (33%)

**For the following CSS rules applying to a `<div id="trans">`, which of the following best describes the effect?**

## Q31. JS variable scope (22%)

**Given the following JavaScript, what is the output?**

## Q32. CSS transform (46%)

**How to rotate an element 45 degrees clockwise?**

## Q33. CSS transform (33%)

**Which of the following is used to move an element on the X-axis by 50 pixels?**

## Q35. DOM structure (33%)

**Given the following HTML, which of the statements is WRONG?**



## Q39. JS naming (41%)

**Which of the following variable names in JavaScript is invalid?**

## Q40. JS naming (63%)

**What will the following JavaScript code alert?**

# Agenda

Mid-term question review (continued from Lec 11)

**Course review**

Programing practice

# Final exam: Sat, May 10<sup>th</sup>, 2:00 – 4:30 PM

## **Coverage:**

All the lectures and labs, except for bootstrap and JS libraries briefly introduced in Lec 11.

## **Form:**

- 8 open-ended questions, each has several sub-questions, involving basic concepts, terminology, HTML/CSS/JS code writing
- Close book, no calculator or scratch paper

# Final exam: FAQ (1)

## 1. How much will it involve code writing?

In some questions, you will be given partial codes and asked to complete the HTML/CSS/JavaScript parts to implement specific design requirements. For example,

Complete the code to create a form as required

Given the HTML code below, complete the CSS parts to modify the element style as required

Given the following HTML structure, complete the JS parts to implement the required interaction features

This kind of questions will take ~ **80%** in the exam (including giving code output)

## 2. How much will the exam cover about JS?

It's hard to give you a specific estimate because oftentimes, the questions combine multiple components (e.g., HTML + JS). If you really want to know the percentage, that will be ~**60%**.

# Final exam: FAQ (2)

## **3. Will there be any multi-choice or multi-answer questions?**

No

## **4. What are other forms of questions than code writing?**

- Most questions will ask you to directly write down the answers. e.g.,
  - What does this tag do?
  - Explain this term in the internet context
  - What will be the output of this code?
  - ...
- ~20% questions asking you to write down the output given a piece of JavaScript code
- ~ 10% for concept/terminology explanation.

# Lec 01 Introduction to the Internet

## **Terminologies in the era of Internet**

- what do the following terms stand for?
  - URL
  - HTTP
  - SMTP
  - SSH
  - Port Number
  - Client/Server network vs. P2P network
  - ...

# Lec 02 – Lec 03 HTML: Basics

## Basic web structure

The arrangement of `<head>`, `<body>`, `<style>`, `<script>`

## Tags to be familiar with

- Basic tags:  
`<p>`, `<h1>`, `<a>`, `<img>`, `<ol>`, `<ul>`, `<li>`, `<div>`, `<br>`, `<button>`
- Advanced tags:  
`<form>`, `<input>`, `<label>`, `<fieldset>`, `<iframe>`, `<video>`
- What are some important attributes of these tags, their possible values, and the ways they affect the element's behavior?



# Lec 02 – Lec 03 HTML: File/URL path

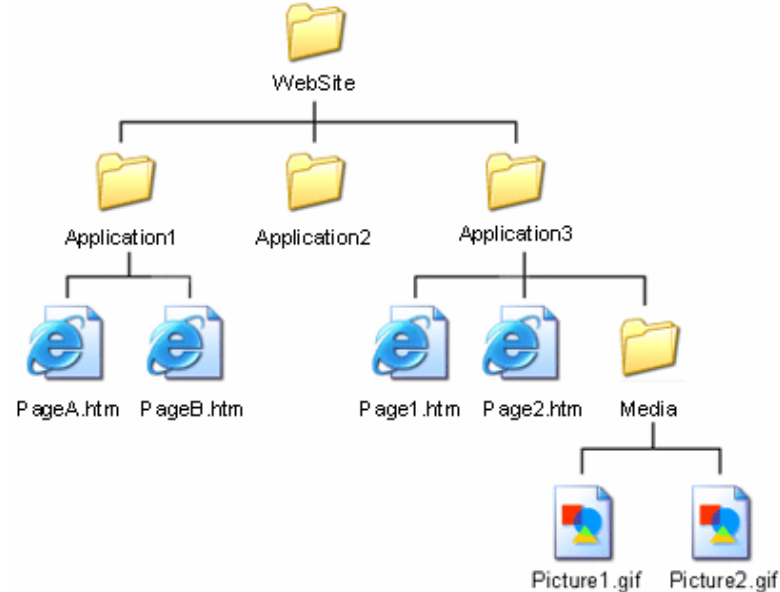
Given a structure of multiple folders and files, how to link each other?

- Relative URL
- Absolute URL

Given the file structure like this ...

- Within `PageA.htm`, how to access `picture1.gif`

With relative URL and absolute URL?



# Lec 02 – Lec 03 HTML: Link & special characters

How to create a link on a webpage, which directs users to other sites?

What if we want to open the link in a new tab?

What if we want the link to direct us to an email address?

What if we want the link to direct us to download a file?

\*What if we want to open another HTML file embedded in a `<div>` of the current webpage? (refer to Lab 5)

## How to handle special characters

- e.g., how to ensure the appropriate display of commonly used special characters, such as `<`, `>`, `©`

# Lec 03 – Lec 04: CSS basics

## Selectors and property modification

- Basic CSS Selectors: tag, id, class, and their differences
- How to link CSS in an HTML and change basic style properties
  - `color`, `background-color`, `font-weight`
- What are some different ways to set the value of colors?
- What are some different ways to link HTML to CSS styles?
  - in-line styling, `<style>` tag, `<link>` tag, `@import`

# Lec 03 – Lec 04: advanced CSS selectors

**DOM structure:** which elements are children and parents, ascendant and descendent?

**Contextual selectors:** what does each of the following selectors select?

- `div p {...}`
- `div > p {...}`
- `div + p {...}`
- `div ~ p {...}`

Combination of basic selectors and contextual selector

- `#container > div {...}`

Pseudo class and pseudo elements

- `p:hover {...}`
- `p:first-letter {...}`

# Lec 05: CSS web elements layout and positioning

## The box model

- What do `margin`, `padding`, `border` means?

## Types of display

- `block`
- `inline-block`
- `none`

## Element positioning

- How to arrange two boxes side by side?
- How to adjust the percentage of width taken by these elements?

# Lec 05-Lec 06: CSS3

## Advanced effects

- transform
  - e.g., how to change the background color of a button when it is hovered?
- transition
  - e.g., how to make the transform happen gradually instead of immediately?
- animation
  - Use of **@keyframes**
    - How to change a button's background color and rotation angle when it is hovered?
    - How to define more than two states of transition?

## Good web dev practice

- How to enhance web accessibility?

# Lec 06 – Lec 07: JS basics

## Expression, operators, and statements

- How to perform basic arithmetic operations?
- What will happen if a string variable is added with a number variable?
- How to append multiple strings?
- How to convert a string into a number?
- The use of `isNaN()`
- Conditionals: the use of `if (...) { ... } else { ... }`
- Loops: how to create for loop?

## Variable scope

- How does variable scope affect the output?
- Differences between `var`, `const`, `let` in declaring a variable

# Lec 08: JS functions

## Declaring and calling a function

- How to declare a function to achieve the required interaction?  
e.g., write a function to print all the elements in an array
- How to call a function when the web page is loaded?

## Use of common built-in functions

- Pop-up windows: `alert(...)` , `prompt(...)`
- DOM objects: how to dynamically display text, image, or video on webpage?
  - `getElementById(...).innerHTML`, `querySelector(...)` , `createElement()`
- Array: how to add, remove elements in different ways?



# Lec 09 – Lec 10: JS objects

## Object creation

- How to create an object?  
e.g., literal, constructor
- What are some commonly used JS built-in objects?
  - `Date()` , `Math`

## Object properties and methods

- How to access an object's properties?
- How to call the methods defined within an object?
- When to use `this`?

# Lec 09 – Lec 10: JS Event Handlers

## Add an event to a web element

- e.g., Display all the elements in an array when the button is clicked
- How to cancel an event?

## Trigger an event periodically

- Differences between `setTimeout (...)` and `setInterval (...)`
- How to cancel such events?

## Good Web Dev practices

- Think about extreme cases where errors may be thrown
- How to catch potential errors in a safer way?

# Lec 09 – Lec 10: JS code efficiency

## Comparing code efficiency of different programs

- When it comes to dynamic content change with `createElement()` and `innerHTML`, which one is more efficient and why?
- JS code usually execute very fast (in a few milliseconds), but if we want to know the efficiency of different codes in achieving the same goal, what can we do?

# Lec 10 – Lec 11: Programing practice

## Extension of array

- e.g., how to dynamically add or remove elements to an array and update the current elements on the webpage?
- If the array elements are not numbers or strings, but are image sources, how to ensure that the images (not their file path) are appropriately shown? (Refer to Lab 8)

## Different ways to add an eventhandler

- `element.onclick=` and `element.addEventListener(...)`
- What are their differences?

## Using JS to control element styling

- `element.style.backgroundColor`, `element.style.fontSize`

# Agenda

Mid-term question review (continued from Lec 11)

Course review

Programing practice

# JavaScript: Array revisit

Complete the function

"displayFruits()" to display all the fruits in the "fruit-container" every time when the page refreshes in a **reversed order**.

```
<script>
fruits = ["apple", "peach", "mango", "grape"];

function displayFruits() {
  var s = "";
```

```
<body>
onload="displayFruits();"
<div id=
"fruit-container"></div>
<button id="add">Add</button> ...
<button id="remove">Remove
(later in, first out)</button> </script>
</body>
```

# JavaScript: Array revisit

Expand the function so that when the fruits are displayed, they are put in an **unordered list**

```
<body
onload="displayFruits();"
<div id=
"fruit-container"></div>
<button id="add">Add</button>
<button id="remove">Remove
(later in, first out)</button>
</body>
```

```
<script>
fruits = ["apple", "peach", "mango", "grape"];

function displayFruits() {
    var s = "<ul>";

    ...

</script>
```

# JavaScript: Array revisit

Expand the function so that the **background color** of each list item will **alternatively change**: the first row is lightgrey, the second row is lightyellow, and the third row is lightgrey, etc

```
<body
onload="displayFruits();"
<div id=
"fruit-container"></div>
<button id="add">Add</button>
<button id="remove">Remove
(later in, first out)</button>
</body>
```

```
<script>
fruits = ["apple", "peach", "mango", "grape"];

function displayFruits() {
    var s = "<ul>";

</script>
```

## Critical thinking:

Are there any other ways to achieve this effect?



# JavaScript: Array revisit

```
<script>  
fruits = ["apple", "peach", "mango", "grape"];
```

```
...  
</script>
```

# JavaScript: `setInterval`

Complete the following JS program that simulates a task reminder system using `setInterval`. The program should remind the user on the **browser console** about a task every minute for five times.

```
let task = prompt("What task would you like to be reminded about?");
```

```
let intervalTime = parseInt(prompt("Enter the interval time in minutes:")) * 60000;
```

# JavaScript: `setInterval`