Learning Code

|  |  |  |
| --- | --- | --- |
| server  season  !GwAPC/-p37Oomkn | 資料夾 p1-20230608 (scarlett)  season\_n1  5\_YmZ.95%D(JRKu | wordpress  season\_n2  YXm9^yJU60#dZbcJWN  @66 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Coursera | edx | github.com | Jsfiddle |  |  |
| 38@ | scarlett38 | Scarlett338 | @66 |  |  |
| Jjk7095 | -nBx)-QB@pi=KQ6 | CuZvqV99rX2j9Cy |  |  |  |

[scarlettliu.job@gmail.com](mailto:scarlettliu.job@gmail.com)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| infinityfree | WP |  | htdocs |  |  |
| @66 | scarlett\_admin | wedding1 | season\_n2 |  |  |
| Ab67016 | kyM9ZL$1&A | 7T17%0NHs@ | YXm9^yJU60#dZbcJWN |  |  |

<https://dash.infinityfree.com/accounts>

https://scarlett.rf.gd

https://scarlett.rf.gd/wp-admin/

http://wedding-kelvin-helen.rf.gd

Administrative URL : http://wedding-kelvin-helen.rf.gd/wp-admin/

## 實用

VS CODE

Node.js

Git > version controlS

Turbo Console Log

Code runner

Indent Rainbow extension

Live Preview extension

JSON

Postman-win64-Setup

# 上網fix

windows+R > devmgmt.msc (上網)

# ●Wordpress

<https://www.youtube.com/watch?v=Z5pMje96Gj0>

How to Create Free WordPress Website 2023 | Get Free Hosting and Domain for WordPress (Proven)

每個platform都有佢自己 file structure，在child theme folder度create files

下載astra，再 child theme

<https://wpastra.com/child-theme-generator/>

Wordpress > appearance > theme > upload

xampp-7.433 > htdocs > 獨立file西 > wp-content > themes > astra // astra-child

(同一個位置)

astra-child > 本來的css, php // assets > css/img/js/lib > css

css / js 我都會放係assets folder

lib 係放其他 jquery既library, 例如 slider, timepicker etc

elementor

# 5-3 version control

<https://www.youtube.com/watch?v=USjZcfj8yxE> <https://www.youtube.com/watch?v=nhNq2kIvi9s> 睇呢兩條video就夠用, 唔洗學得好深入

重點

Distributed Version Control System – git, Mercurial,

Centralised Version Control system, for other developers to see your changes you must commit them to a central server.

?

//Connecting to GitHub via SSH

/\* ssh-keygen -t ed25519 -C "your@email.com"

Generating public/private ed25519 key pair.

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

Your identification has been saved in ./ssh/private\_key.

Your public key has been saved in ./ssh/public\_key.pub.

The key fingerprint is:

SHA256:UDQI5N1FL3QSq7Gj1o12mkr9Me7qGMZAeE1s9BWIln4 your@email.com

The key's randomart image is:

+--[ED25519 256]--+

|   .o+o=+oOo.    |

|   o +B+.= =     |

|  . +++ + o .    |

|   o  ..E+ .     |

|    .  .S        |

|     o + +       |

|      B = =      |

|     + + \* o     |

|      oo=o+      |

+----[SHA256]-----+

 \*/

●The Command line

GUI applications

cmd

windows+R > devmgmt.msc (上網)

轉位 cd

開新夾 mkdir

開新file type nul > 或 echo.>

開啟檔 code

show ls

顯路徑 pwd

copy cp

move mv

remove rm

再黎過cmd clear

~ 英文tilde

|  |  |  |
| --- | --- | --- |
| change directory  轉而家location | cd | cd ~/Desktop 去桌面  cd .. 向上  cd / |
| 開新file | type nul > | type nul > your\_file.txt |
|  | > | ls -1 > hi.txt |
|  | echo.> | echo.> your\_file.txt |
|  | call >> | call >> your\_file.txt |
|  | touch | touch example.txt |
| 開新folder (make directory | mkdir | mkdir word-docs |
| open file | code | code word.txt |
| show contents | ls | ls –l 開list order file  ls –a 開所有 |
| 顯路print working directory | pwd |  |
| copy | cp |  |
| move | mv | mv a b  move a入b |
| 再黎過cmd | clear |  |
|  |  |  |
|  |  |  |
| view history | history |  |
| 手冊 | man | man ls |
| 閱讀file  Allows reading or concatenation | cat |  |
| Displays the contents of a file one page at a time. | less |  |
| 搵野  Global regular expression, allows for searching contents of files or folders | grep  grep –i 頭尾 ??  grep –w 完全一樣 | grep Sam names.txt  case sensitive, 只出 words that begin with “Sam”. |
| used for removing a file or a directory | rm |  |
| 有幾多字 | wc hi.txt -w |  |
| folder內有幾多file | ls | wc -w |  |
| pipes to combine commands togethe | | |  |
| Redirection |  |  |
| standard input  <  cat | cat < hi.txt  cat > hi.txt  “文字”  ctrl + d (完) |  |
| ouput  > | > |  |
| error | 2>  2>&1 | 2> error.txt |
|  |  |  |

What are Unix commands?

Unix Linux

Using Bash on Windows

less .bashrc

.bashrc

input = 0

output =1

error=2

----------------

Version control

As you know by now, version control is a system that records changes to a file or set of files over time so that you can access specific versions later. In software development, **Version Control Systems (VCS)** allows developers to manage changes to their code and track who made each change. But how did this software come about?

Version Control has a long history going back to the 1980s. In fact, version control systems were created before the Internet!

One of the first significant Version Control Systems was the **Concurrent Versions System (CVS)**. It was first developed in 1986 by Walter F. Tichy at Purdue University and released publicly in 1990.

**CVS** stores information about every file in a folder structure, including the name of the file, its location in the folder structure, who last modified it, and when it was last modified. The **CVS** also stores information about folders, including their names and who created them.

It was popular for many years; however, it has some significant flaws in its design. **CVS** does not include integrity checks which means your data can become corrupted. When you update or submit changes to the system, if an error occurs, the system accepts the partial or corrupted files. Additionally, the system was designed mainly for text files, not binary files such as images or videos.

The main successor to **CVS** was **Subversion (SVN)**.

CollabNet developed Subversion in 2000 and solved many of the issues present in **CVS**. To ensure data integrity, it included integrity checks in its design. It also supported the versioning of binary files better than **CVS**. Thanks to these improvements, SVN became popular in the open-source community with free hosting being offered for open-source projects by Google and SourceForge.

However, Subversion used a centralized **VCS** model. This means that all operations have to be done using a centralized server. If the server were down or slow, this would impede development.

In 2005, two new projects were started to develop distributed version control systems; Mercurial and Git. Both projects were created in response to an event involving the Linux kernel development.

Previously, the Linux kernel was using a proprietary **VCS** known as BitKeeper. BitKeeper was one of the first distributed version control systems initially released in 2000. BitKeeper had originally provided a free license to Linus Torvalds to support Linux’s development. However, in 2005, the license was revoked. This controversy led to the creation of the Mercurial and Git projects.

Mercurial was developed by Olivia Mackal. It is developed as a high-performance distributed **VCS**. Many platforms offering Subversion hosting began to offer Mercurial hosting too. It became popular as Subversion users found it easy to transition to a Mercurial repository, thanks to the hosting providers and its small learning curve.

Git was developed by Linus Torvalds to host the Linux kernel’s source code. Like Mercurial, it is a distributed **VCS**. Its first public release came in 2007.

Git became popular in the open-source community due to its distributed **VCS** design and Github offering free Git hosting for open-source projects. Git has since become the selected version control system for many open-source and proprietary software projects.

Version control in professional software development

Version Control plays a crucial part in software development. As a developer, you’ll work with other developers on projects to deliver software to customers. Depending on the role, you could be working with a small team of 2 or 3 developers in a single project or a large team spanning multiple projects. In either scenario, Version Control will be a crucial tool to help your team succeed.

However, Version Control must be complemented by other tools and procedures to ensure quality and efficiency throughout the software development process. In this lesson, we’ll explore some of the common tools and strategies developers use in conjunction with Version Control.

Workflow

Using Version Control without a proper workflow is like building a city without traffic lights; without appropriate management, everything will turn into chaos.

For example, let’s say you’re working on a big project and editing a file. Another developer also starts editing a file. Both of you submit the file to the VCS at the same time. Now there’s a conflict! How should the conflict be resolved? A good workflow will have a process for resolving conflicts.

Another example is when a new junior developer is joining your team. If the project code is used for a critical system, it is risky to allow them to submit code changes directly. To solve this, many developers use a peer review system where another developer must review code before it can be merged in.

Workflows are essential to ensure code is managed correctly and reduce mistakes from happening. Different projects will have different workflows. In this course, you’ll learn some common workflows using the Git Version Control System.

Continuous Integration

Continuous Integration, or CI, is used to automate the integration of code changes from multiple developers into a single main stream. Using a workflow whereby small changes are merged frequently, often many times per day, will reduce the number of merge conflicts.

This process is widespread in test-driven software development strategies. CI is often used to automatically compile the project and run tests on every code change to ensure that the build remains stable and prevent regressions in functionality.

Continuous Delivery

Continuous Delivery is an extension of Continuous Integration. Once the changes have been merged into the main stream, a Continuous Delivery system automatically packages the application and prepares it for deployment. This helps avoid human error when packaging the application.

Continuous Deployment

Continuous Deployment is an extension of Continuous Delivery. The goal of Continuous Deployment is to deploy and release software to customers frequently and safely. The strategy commonly involves automatically deploying to a test (also known as staging) environment first to validate the deployment package and software changes. Once validated, it can automatically deploy to the live (also known as production) environment for customers.

Conclusion

With these tools and procedures, it is possible to understand how software starts from a developer writing code to being deployed live for customers to use. Of course, there is much more to running a live software service, but that is a lesson for another day.

Staging vs. Production

Development Environments

Every development team prior to releasing their new features or changes needs to verify that the code they do release is not going to cause any issues or bugs. In order to achieve this, they normally set up multiple environments for different ways to test and verify. A common practice is for teams to have a developer environment, a UAT or QA environment, and a staging environment. The main purpose of this flow is to find any potential issues that may arise due to changes or new features being added to the codebase. The more ways to test the changes the less likely bugs will be introduced.

Staging

The staging environment should mimic your production environment. The reason for this is because you want to test the code in an environment that matches what you have in production. This allows teams to spot or find any potential issues prior to them getting to production. The closer the staging environment is to your production, the more accurate your testing is going to be. Staging environments can also be used for testing and verifying new features and allow other teams including QA or stakeholders to see and use those features as a pre-trial. Staging should also cover all areas of the architecture of the application including the database and any other services that may be required. Areas that benefit from staging environments include:

New Features

Developers submitting new features along with feature flags for turning them on and off should always do a testing round in a staging environment. They allow teams to verify that the feature works, it can be turned on and off via configuration flags and also that it does not break or interfere with existing functionality.

Testing

As the staging environment mimics your production environment, it's also a great place to run tests. QA teams will normally use it to verify new features, configuration changes or software updates/patching. The types of testing covered will be Unit testing, Integration testing and performance testing. All except performance testing can also be carried out in production. Performance can also be completed in production but only at specific times - usually out of hours as it will have a drastic effect on the user experience.

Sometimes it is not always feasible to have an exact replication either due to costs or time. Certain areas can be cut back - for example, if your service is load balanced on 10 virtual machines in production, you could still have 4 virtual machines in staging. The underlying architecture is the same but the overall performance may be different.

Migrations

Staging is a perfect place to test and verify data migrations. Snapshots can be taken from production and used to test your migration scripts to confirm your changes will not break anything. If in the case it does cause an issue, you simply rollback and try again. Doing something like a migration in production is extremely risky and error-prone.

Configuration Changes

Configuration can also cause headaches for teams, especially in a large cloud-based architecture. Having a staging environment will allow you to spot any potential issues or bottlenecks.

Production

Production is live. It's out there for people to see and/or interact with. Any issues or problems you may have had should have been caught and fixed in the staging environment. The staging area gives the team a safety net to catch these possible issues. Any code that is deployed to production should have been tested and verified before the deployment itself.

Downtime

Downtime for any service especially customer facing will most likely be revenue impacting. If customers can not access or use your website or app to its full capabilities, it will most likely have a cost involved. Take for example an e-commerce company that allows users to buy goods and services online. If they release a new feature to their shopping cart which actually breaks the payment process, this will have an impact on customers not being able to buy goods online.

Vulnerabilities

Cyber-security should also play a big role in what gets released in production. Any updates to software such as patching or moving to the latest version should be checked and verified. This is also the same rule for not upgrading software when critical updates are released.

Reputation

Downtime or issues in production is damaging for a company as it does not instill confidence in end users. If something is down or broken it can cause the company to lose potential customers.

# 5-4

HTML and CSS in depth

Week 2

Debugging the front-end

In this reading, you will learn about some of the fundamental tasks used to debug CSS. The scope of what’s covered will match the topics already covered in this course.

You may have come across websites that have misaligned or overflowing text, broken web links or websites that take too much time to load. While the front-end and back-end both contribute to faulty web pages, visible styling is primarily concerned with the front-end.

CSS is a styling language, unlike conventional programming languages such as Python and Java, that has controls and follows a logical structure. This can make it difficult to find the exact root of the problem. Additionally, as you know, CSS does not flag errors, and most of the ‘bugs’ that you see in CSS are aesthetic in nature and need human intervention to solve. The task of debugging the front-end is more about experience than knowledge.

The first step in debugging CSS is to find the root of the issue and isolate the elements involved. The majority of CSS issues will be with the layout, such as:

Content overflow from parent to child or container class

Misplaced elements in relation to its container class

Browser and device-related inconsistencies resulting in variable viewports

Isolation by reduced test case

One way to deal with a problem is to replicate your code and systematically remove any code unrelated to the HTML and CSS elements that could be the source of the problem. The code should be distilled down to the least amount of code possible, and only then are suitable changes made to get the results you want. Alternatively, you can enable rules one at a time to observe their impact on the displayed elements.

Items inside containers

Often, isolation will not work, as the problem is the result of the relative mapping of elements. For example, with the misconfigured width of an item inside a flex layout. It’s important to check the use of suitable CSS properties in such cases. For a given item inside a grid, depending on the use case, width, grid-template-column, margin and padding can all be used to give spacing to an element. Additionally, you can also set different units that will all have their own behavior. In most cases, it helps to be familiar with the unit of measurement in relation to the container type to avoid misconfigurations.

Relocating items

Similar to the isolation of elements, you can move a certain element to observe its behavior. Doing a comparison can often help you to understand the source of the problem.

The CSS compiler reads the elements from right to left. As an example, for a selector such as div .alpha > p, the element read first will be p before moving ‘outside’. When you change the position of the p from inside the .alpha class to some other position inside your code, it is easier to debug the source of the problem. This should be done on a case-specific basis.

Getting familiar with the box model

The box model is a very powerful source of information and can solve many issues with alignment. Using margin, padding and border is useful, but can be tricky and must be well understood.

Browser issues

Many times, the styling you have renders correctly in the IDE but misbehaves in a browser. That is because browsers have their own default CSS stylesheets called user-agent styles. While modern-day browsers are mostly compatible, you may encounter minor inconsistencies. Overriding the browser's settings can be done with universal selectors, in such cases written at the top of the code, and will include properties such as ‘margin: 0px;’ to reset the margin values set by the browser by default.

Dev tools

There are lots of user-friendly tools available today that can help debug CSS. However, the best tool you can use is the one provided by the browsers, called the developer tools, or dev tools. You can find these by right-clicking on a web page and selecting ‘Inspect Element’. option Note how every browser has its own expression when it comes to the configuration options, but fundamentally they are similar. Browsers today are very powerful pieces of software. If you spend time exploring the options, you may not need any other additional tools or software for debugging CSS and other front-end languages.

A couple of the important options you can find inside your browser include:

Sources: Lists the filenames such as HTML, and CSS files used by the webpage that can be explored

Elements: Scrolls through the code to select a specific element for exploration

Inside the Elements tab, on the right-hand side, you will find several options such as Computed layouts that will show the box model, Layouts that contain page and grid overlay options, and Font.

You can select a specific element much more easily with the help of the ‘Element selection’ icon inside the dev tools. It enables the selection of specific elements on your web page. You can also access this option by hovering over a specific element on your page that will display its properties to you.

Hovering Icon

Targetting Icon

On selecting the desired element, one way to use the Elements tab for debugging is by right-clicking on that element, scrolling to ‘Copy’ and then selecting an option from the drop-down list that appears.

Display of submenu from copy option

This way, you can explore the isolated code to find the problem.

Additionally, you can add the Breakpoint option that is more useful for interactive pages by selecting the ‘Break on’ option and selecting an option from the dropdown list that appears.

Break on Submenu options

Nowadays, some browsers are providing options such as Cumulative Layout Shift (CLS) that helps determine the overall efficiency of a web page.

You can also bring up the element families by clicking on the horizontal bar:

Horizontal bar and element families

One very useful feature is the ability to make changes in your code directly inside the browser. First, use the ‘element selector’ mentioned above to select some elements and then look for the ‘+’ icon inside the dev tools. You can directly start adding relevant rules for that particular element which will immediately display changes on your web page. The changes you make can also be tracked from the ‘Changes’ tab. The live interactive nature of this feature greatly improves the experience of debugging.

‘Console’ is another feature that, although it is not that useful in CSS, will come in handy as you deal with active or dynamic web pages along your developer journey.

There are also responsive design modes in CSS that allow you to render your webpage to a specific browser or device. In addition to these, there are numerous ways in which you can explore and configure settings inside dev tools.

When it comes to designing and styling CSS, if you don’t understand how it works, all issues will appear to be bugs. If you look at the fundamental structure of CSS, it consists of rulesets containing selectors and declaration blocks that contain properties and values. Micro-assessment of formatting and its validity can be done to troubleshoot the source of the problem. Practicing the creation of web pages and exploring the dev tools is the best way to get better at debugging and CSS in general.

In this reading, you learned about some of the fundamental tasks used to debug CSS.

Creating themes

互補色非常適合包含條形圖和圖表的網站。它還取決於其他幾個因素，例如網頁所需的顏色數量、網頁的主題和領域、讀者的用戶人口統計等。例如，醫療服務信息網站通常具有白色背景。除了提到的配色方案之外，如果造型和設計得到很好的呈現，即使是只有黑白的非彩色主題也可以很有吸引力。

不平衡的網頁設計可能會導致我們所說的認知超載。認知過載是指向用戶呈現過多的信息或活動，這可能會損害用戶體驗。

受自然啟發的色調幾乎被普遍認為更令人愉悅並增加了審美價值。隨著人們花在互聯網上的時間越來越多，一個好的主題可能會增加或破壞網頁的吸引力。通常，最好在手邊保留一種以上顏色組合的調色板，以便您可以在網頁上進行實驗。

具有適當顏色組合的簡單主題可以通過創建視覺層次結構來幫助用戶導航網站。例如，如果用戶觀察副標題的特定配色方案，他們將能夠了解在網頁上何處查找某些信息。這意味著您可以避免過多的設計元素，例如換行符和網格，因為它們不會增加網頁的吸引力。

The combination of colors used in the design of a website is called the color scheme. Colors are one of the most important components of any website today because it sets the tone for the viewer. For the optimal use of colors, web developers make use of color theory and color schemes.

Color schemes

Fundamentally all colors are a combination of the three colors: red, yellow, and blue. These are called the primary colors. Secondary colors like orange, purple and green are a combination of two primary colors. Finally, there are six tertiary colors formed from a combination of the three primaries with secondary colors. By adding more black or more whiteyou can create lighter or darker tints of each color. All of these colors and tints make upa color wheel of 12 colors. This color wheel is the foundation for any color scheme.

Depending on the relative positions of colors on the wheel, there are seven commonly identified color schemes according to color theory that can be used for maximum appeal. These are:

· Monochromatic

· Analogous

· Complementary

· Split complimentary

· Triadic

· Square and

· Rectangle

The different combinations are represented in the image below.

Different color combinations on the color wheel that are based on the relative position of colors to others.

Image source:

https://www.moving.com/tips/how-to-choose-a-color-scheme-for-your-home/

Using the color schemes

Each of the different color schemes are used with the purpose of enhancing a website. While you can pick any of the different schemes, a practiced UX designer will considerwhether a specific type of color scheme is more suitable for certain websites than others. For example, complementary colors are well-suited for websites that will have bars and charts. It also depends on several other factors such as the number of colors that you are going to need for your webpage, the topic and domain of the webpage, user demographics of the readers and so on. For example, an informational website for medical services will typically have a white background. Other than the color schemes mentioned, even achromatic themes with only black and white color can be appealing if styling and design is well presented.

Factors to consider for themes and colors

While color schemes serve as a guide for picking website colors, there are some factors that must be considered in the process. The user experience must be of the utmost priority as a good webpage must be able to sustain the attention of the reader withoutoverwhelming them. A good webpage will have a good balance of information and design elements. An unbalanced web design can lead to what we call as cognitive overload. Cognitive overload is when too much information or activity is presented to the user which may be detrimental to the user experience.

There is a field of research in psychology that focuses on the effects color has on the mood of a user. Nature-inspired color tones are almost universally seen to be more pleasant and add aesthetic value. As people are increasingly spending time on the internet, a good theme may make or break the appeal of a webpage. Often, it is a good idea to keep a palette of more than one color combination on hand so you can experiment with your webpage.

A color theme on a website also plays a role in other aspects beyond the aesthetics and help in shaping the user experience of the website. A simple theme with appropriate color combinations can help a user navigate the website by creating a visual hierarchy. If for example, a user observes a specific color scheme for the subheadings, they will be able to understandwhere to look for certain information on the web page. This means you can avoid excessive design elements such as line-breaks and grids where they are not adding to the appeal of the web page.

While the experience of choosing the right colors for your website is a process of trial and error, inspiration can be drawn from a webpage that has held a personal appeal. Color theory is an important aspect of the user experience domain and plays an important role in website design so don’t be afraid to experiment with different color schemes.

W1- Input types

Submit

Displays a submit button for submitting all values from an HTML form to a form-handler, typically a server. The form-handler is specified in the form’s "action" attribute:

<input type="submit" value="Submit" />

Checkbox

<input type="checkbox" id="dog" name="dog" value="Dog">

<label for="dog">I like dogs</label>

<input type="checkbox" id="cat" name="cat" value="Cat">

<label for="cat">I like cats</label>

Radio

<input type="radio" id="light" name="theme" value="Light">

<label for="light">Light</label>

<input type="radio" id="dark" name="theme" value="Dark">

<label for="dark">Dark</label>

Password

Defines a single-line text field whose value is obscured, suited for sensitive information like passwords.

<label for="pwd">Password:</label>

<input type="password" id="pwd" name="pwd">

Date

Displays a control for entering a date with no time (year, month and day).

<label for="dob">Date of birth:</label>

<input type="date" id="dob" name="date of birth">

Datetime-local

Defines a control for entering a date and time, including the year, month and day, as well as the time in hours and minutes.

<label for="birthdaytime">Birthday (date and time):</label>

<input type="datetime-local" id="birthdaytime" name="birthdaytime">

File

Displays a control that lets the user select and upload a file from their computer. To define the types of files permissible you can use the "accept" attribute. Also, to enable multiple files to be selected, add the "multiple" attribute.

<label for="myfile">Select a file:</label>

<input type="file" id="myfile" name="myfile">

Hidden

Defines a control that is not displayed but whose value is still submitted to the server.

<input type="hidden" id="custId" name="custId" value="3487">

Image

Defines an image as a graphical submit button. You should use the “src” attribute to point to the location of your image file.

<input type="image"src="submit\_img.png" alt="Submit" width="48" height="48">

Number

Defines a control for entering a number. You can use attributes to specify restrictions, such as min and max values allowed, number intervals or a default value.

<input type="number" id="quantity" name="quantity" min="1" max="5">

Range

Displays a range widget for specifying a number between two values. The precise value, however, is not considered important. This is typically represented using a slider or dial control. To define the range of acceptable values, use the “min” and “max” properties.

<label for="volume">Volume:</label>

<input type="range" id="volume" name="volume" min="0" max="10">

Reset

Displays a button that resets the contents of the form to their default values.

<input type="reset">

Search

Defines a text field for entering a search query. These are functionally identical to text inputs, but may be styled differently depending on the browser.

<label for="gsearch">Search in Google:</label>

<input type="search" id="gsearch" name="gsearch">

Time

Displays a control for entering a time value in hours and minutes, with no time zone.

<label for="appt">Select a time:</label>

<input type="time" id="appt" name="appt">

Tel

Defines a control for entering a telephone number. Browsers that do not support “tel” fall back to standard text input. You can optionally use the "pattern" field to perform validation.

<label for="phone">Enter your phone number:</label>

<input type="tel" id="phone" name="phone" pattern="[+]{1}[0-9]{11,14}">

Url

Displays a field for entering a text URL. It works similar to a text input, but performs automatic validation before being submitted to the server.

<label for="homepage">Add your homepage:</label>

<input type="url" id="homepage" name="homepage">

Week

Defines a control for entering a date consisting of a week-year number and a year, with no time zone. Keep in mind that this is a newer type that is not supported by all the browsers.

<label for="week">Select a week:</label>

<input type="week" id="week" name="week">

Month

Displays a control for entering a month and year, with no time zone. Keep in mind that this is a newer type that is not supported by all the browsers.

<label for="bdaymonth">Birthday (month and year):</label>

<input type="month" id="bdaymonth" name="bdaymonth" min="1930-01" value="2000-01">

Button

This displays a clickable button and it’s mostly used in HTML forms to activate a script when clicked. <input type="button" value="Click me" onclick="msg()" />

Keep in mind you can also define buttons with the <button> tag, with the added benefit of being able to place content like text or images inside the tag.

<button onclick="alert('Are you sure you want to continue?')">

<img src="https://yourserver.com/button\_img.jpg"

alt="Submit the form" height="64" width="64">

 </button>

Cheat sheet: Interactive form elements

When filling in HTML forms, we expect users to abide by certain rules, like using numbers when asked to, or properly formatting a URL or an email when needed. However, humans are prone to errors and in some cases, they may overlook some of the data they input. That’s why it’s important to ensure the shape of the data we expect in each field is correct. HTML form validation is a set of attributes we can add to form inputs to perform automatic validation on the user’s behalf. The most important attributes you’ll find yourself using for validation are the following.

Required

Denotes a mandatory input that the user can’t leave empty. It can be used with any input type, like password, radio, text and so on.

<input type="text" id="firstName" name="firstName" required>

Maxlength

Specifies the maximum length of a text input, in other words, the maximum number of characters that can be entered for a specific field. If provided, it will prevent the user from entering more characters than the limit.

<input type="text" id="description" name="description" maxlength="50">

Minlength

Specifies the minimum length of a text input. If set, the input will not accept fewer characters than those specified.

<input type="password" id="password" name="password" minlength="8">

Min and max attributes

Determine the minimum and maximum values allowed for an input field. They are usually applied to numerical text inputs, range inputs or dates.

<input type="number" id="quantity" name="quantity" min="1" max="10">

<input type="range" id="volume" name="volume" min="1" max="100">

Multiple

Indicates that the user can enter more than one value in a single input field. This attribute can only be used for email and file input types.

<input type="file" id="gallery" name="gallery" multiple>

Pattern

Defines a particular pattern that an input field value has to fulfill to be considered valid. This attribute expects a regular expression to specify the pattern. It works with text, date, search, URL, tel, email and password input types. For example, you can restrict phone numbers to be only from the UK.

<input type="tel" id="phone" name="phone" pattern=”^(?:0|\+?44)(?:\d\s?){9,10}$” >

Grids and flexbox cheat sheet

Note: ‘|’ stands for alternatives or OR.

Grid

The syntax for creating a grid:

123

selector{

display: grid; /\* or inline-grid \*/

}

Grid shorthand consists of the following properties with default values:

grid

A grid will allow you organize the various elements on your page.

grid-template-rows: none

This feature allows you configure your elements so that they are organized similarly to rows on a table.

grid-template-columns: none

This feature allows you configure your elements but with this setting the elements are organized like columns on a table.

grid-template-areas: none

This feature allows you configure the names of a grid and how they sit in relation to one another.

grid-auto-rows: auto

Default setting for all row sizes that have not been explicitly configured.

grid-auto-columns: auto

Default setting for all column sizes that have not been explicitly configured.

grid-auto-flow: row

Default location for rows that are not explicitly allocated.

column-gap: normal

This sets the gap between the columns

row-gap: normal

This sets the gap between the rows

Grid properties for container

grid-template-columns: measurement units | % units |repeat()

Defines the line names, and maintains a constant size of column items. Can accept a range of different measurement sizes.

grid-template-rows: measurement units | % units |repeat()

Defines the line names, and maintains a constant size of rows. Can accept a range of different measurement sizes.

grid-auto-columns: measurement unit (fixed value for all columns)

Determines the default size for columns that have not been explicitly configured.

grid-auto-rows: measurement unit (fixed value for all rows)

Determines the default size for rows that have not been explicitly configured.

grid-template: “header header” auto

This allows you define and maintain named cells on a grid

“main right” 75vh

This defines two cells named main and right, that have a sizing of 75% of the viewport height.

“footer footer” 20rem

This defines two cells named footer and footer, that have a sizing of 20 root em (rem). This defines the size in relation to the html font size.

Gap

grid-gap: measurement units

Determines the gap between rows and columns

grid-column-gap: measurement units

Determines the gap between columns

grid-row-gap: m-unit-1 m-unit-2

Determines the gap between columns

Alignment

justify-items: start | center | end | stretch

Defines the default space that is allot to each item on the grid

align-items: start | center | end | stretch

Defines the default space related to an item along the grid’s block axis

place-items: start | stretch /\* shorthand for two properties above \*/

This feature allows you align items with the block and inline directions.

Justification

justify-content: start | center | end | stretch | space-between | space-evenly | space-around

Defines browser allocation of space to content items in relation to the main-axis

align-content: start | center | end | stretch | space-between | space-evenly | space-around

Defines browser allocation of space to content items in relation to cross axis and block axis

place-content: center | start

This feature allows you align items with the block and inline directions.

Positioning

grid-auto-flow: row | column | dense

This relates to how the items are placed automatically within the grid

grid-auto-columns: measurement units

This relates to the size for columns created without specific size specifications

grid-auto-rows: measurement units

This relates to the size for rows created without specific size specifications

Grid properties for items (child)

grid-column: column position /\* E.g. 1/2 \*/

Allows for specifying where on the grid the column is to start.

grid-column-start: column start position

This property determines the starting column position an item is placed on a grid.

grid-column-end: column end position

This property determines the end column position an item is placed on a grid.

grid-row: row position /\* E.g. 1/2 \*/

Allows for specifying where on the grid the row is to start.

grid-row-start: row start position

This property determines the starting row position an item is placed on a grid.

grid-row-end: row end position

This property determines the end row position an item is placed on a grid.

Justification and alignment

justify-self: start | center | end | stretch

Determines how an item is positioned inside its aligned container in relation to the appropriate axis.

align-self: start | center | end | stretch

Aligns an item within a grid area.

place-self: start | stretch /\* shorthand for two properties above \*/

This setting lets one align and justify an item within a block.

Flexbox

The syntax for creating a flexbox:

1234

selector{

display: flex | inline-flex

}

Here the selector can refer to any of the following flex attributes

Attribute selector

Class Selector

ID Selector

Type Selectors

Universal Selectors

The display relates to how you want the selector to be shown. Setting display to flex makes the given selector a flex box. Setting display to inline-flex makes the selector a flex box container while will be inline.

Properties for flexbox container

flex-direction: row | row-reverse | column | column-reverse

It is possible to specify the direction your elements will follow. Traditionally text goes from left to right which is flex’s default setting however it can be set from right to left or even top to bottom. The four flex-direction are:

row : organized from left to right

row-reverse: organized from right to left

column: organized from top to bottom

column-reverse: organized from bottom to top.

flex-wrap: wrap | nowrap

The standard layout is to plot the elements from left to right in a straight line. The wrap feature allows you customize this to match the size of the window displaying the page.

wrap: Automatically wrap the items with as the window space gets smaller.

Nowrap: Default setting, items remain rigid and don’t respond to adjustments made to the window size.

align-items: flex-start | flex-end | center |Stretch

This determines how the flex items are to be positioned on the page. Items can be aligned in a variety of ways

Flex-start: Similar to standard writing, items start at the top left-hand corner and are positioned from left to right

Flex-end: Position begins in the bottom right hand corner.

Center: Item is positioned from the center.

Stretch: item expands to fill the container.

justify-content: flex-start | flex-end | center | space-between | space-evenly

Justify-content determines the alignment of the flex items.

Flex-start: goes from right to left along the main axis.

Flex-end: goes from left to right along the main axis.

Center: Starting at the middle, alignments expands from there.

Space-between: first and last item are flush with the left and right wall respectively, every other item is evenly spaced.

Space-evenly: each item is equidistant from each other and the boundary wall

Properties for flexbox items (child)

flex-grow: factor of flex’s main size

This attribute enables the flex container to grow proportionally to the other containers present.

flex-shrink: factor of flex’s main size

This allows elements to shrink in relation to items around it.

flex-basis: auto | factor of main’s size | measurement unit

The sets the initial main size of an item. It can be overridden if other stylized elements are configured.

order:position in flex /\* Set ascending by default \*/

The standard positioning of items is by source order, however this feature will enable you to configure where the items appear on the page.

align-self: start | center | end | stretch

This determines where on the page the child items will be positioned. Similar to the main flex attributes, start is to the left and end is to the right.

# 初期如何學習寫程式？

* 步驟

1. 大概想一下解法，不寫程式碼
2. 把解法寫成 pseudo code（虛擬碼）
3. 把 pseudo code 翻譯成程式碼

* 利用條列式寫法

1. 將大問題分割成小問題
2. 一行只做一件事
3. 善用敘述、條件判斷
4. 善用跳轉（jump）來實現重複執行

# ●VS 快鍵

Alt + shift

|  |  |
| --- | --- |
| 複製向下 | Alt + shift + down arrow |
| 快速註解/取消註解 | alt+shift+A |
| 調位 | alt+上/下 |
| 成句highlight | shift+home |

# template

<https://htmlcheatsheet.com/css/>

HTML

<!doctype html>

<html lang="en">

    <head>

        <meta charset="UTF-8" />

        <meta http-equiv="X-UA-Compatible" content="IE=edge">

        <meta name="viewport" content="width=device-width, initial-scale=1.0">

        <title> </title>

        <link href="bootstrap.min.css" rel="stylesheet">

        <link href="自己.css" rel="stylesheet"> <!-- 自己 -->

        <style>

        </style>

    </head>

<body>

    <header></header>

    <H1></H1>

<!---------------------------------------------------------- script -->

    <script src="jquery-3.7.0.js"></script> <!-- 未知 -->

    <script src="bootstrap.min.js"></script>

    <script src="自己.js"></script> <!-- 自己 -->

    <footer></footer>

</body>

</html>

CSS

\* {

    box-sizing: border-box;

    margin:0;

    padding:0;

  }

body {

    font-size: 16px;

    color

    background-color

    font-family

    margin: 0;

    margin-top: 0;

    margin-bottom: 0;

    padding:0;

}

body h1, body h2, body h3, body h4, body h5, body p{

  margin-top: 0;margin-bottom: 0;

}

\* {

    box-sizing: border-box;

    margin:0;

    padding:0;

  }

body {

    font-size: 16px;

    color

    background-color

    font-family

    margin: 0;

    margin-top: 0;

    margin-bottom: 0;

    padding:0;

}

body h1, body h2, body h3, body h4, body h5, body p{

  margin-top: 0;margin-bottom: 0;

}

/\* 網上temp \*/

html,body,div,span,applet,object,iframe,h1,h2,h3,h4,h5,h6,p,blockquote,pre,a,abbr,acronym,address,big,cite,code,del,dfn,em,img,ins,kbd,q,s,samp,small,strike,strong,sub,sup,tt,var,b,u,i,center,dl,dt,dd,ol,ul,li,fieldset,form,label,legend,table,caption,tbody,tfoot,thead,tr,th,td,article,aside,canvas,details,embed,figure,figcaption,footer,header,hgroup,menu,nav,output,ruby,section,summary,time,mark,audio,video {

  margin: 0;

  padding: 0;

  border: 0;

  font-size: 100%;

  font: inherit; vertical-align: baseline;

  }

  article,aside,details,figcaption,figure,footer,header,hgroup,menu,nav,section {

  display: block;

  }

  body {line-height: 1;}

  ol,ul { list-style: none;}

  blockquote,q { quotes: none;}

  blockquote:before,blockquote:after,q:before,q:after { content: ''; content: none;}

  table { border-collapse: collapse; border-spacing: 0;}

# ============

# Html css Notes

雜 (NOTES)

<https://placehold.co/>

max-width: 80%;

By default, a block-level element tries to take up as much horizontal space as its containing element will allow

a:hover div:nth-child(5) {

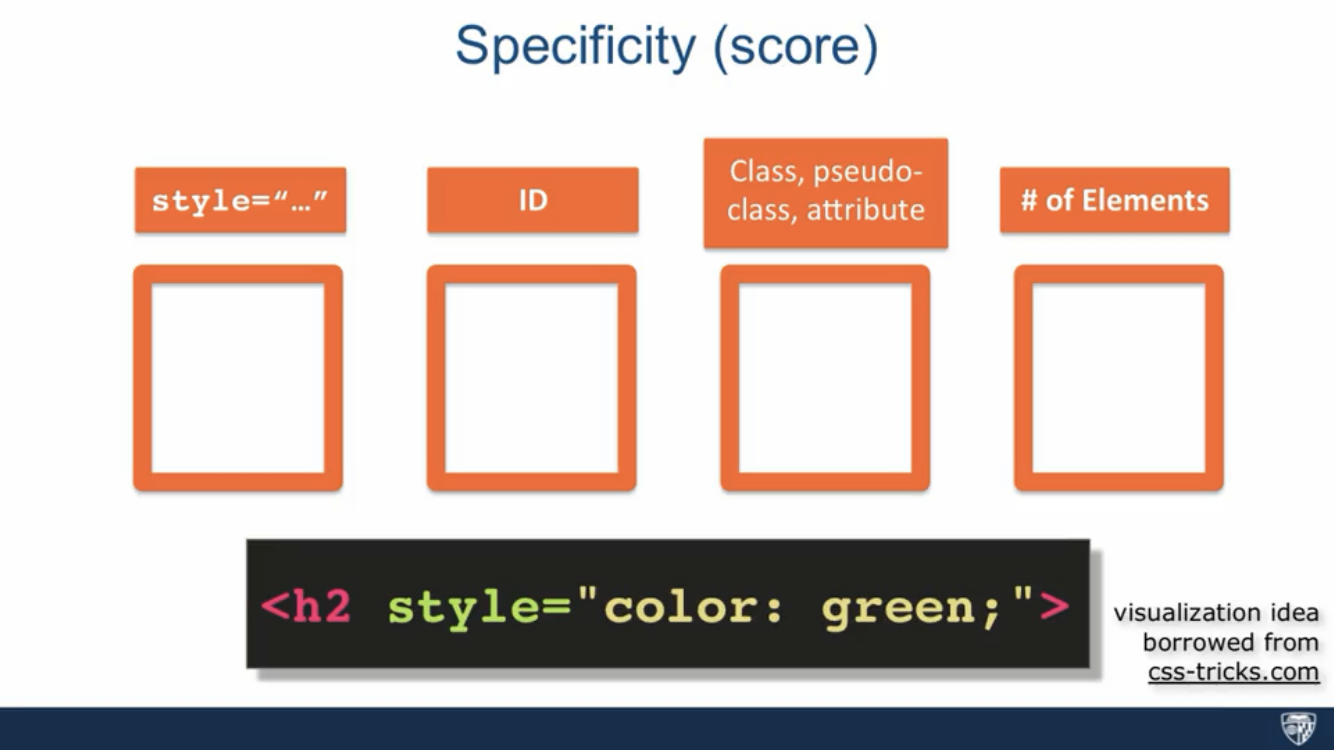
font-size: 24px;

}

What color will the words "I am having a color identity crisis again! Help!" be?

An ID selector has a higher specificity score than the class selector, so it will win.

CSS selector



|  |  |  |
| --- | --- | --- |
|  | 會影響 | X |
| p.big  所有p has c.big | <p class="big"></p> | <div class="big"></div> |
| article > p  所有親生仔 | <article><p></p></article> | <p></p>  <article><div><p></p></div></article> |
| article > .red  art內所有親生仔with .red class | <article>      <p class="color"></p>      <div>          <p class="color"></p> 冇轉      </div>  </article> |  |
| article p  所有p子孫 | <article><p></p></article>  <article><div><p></p></div></article> | <p></p> |
| .red p  .red所有p子孫 |  |  |

|  |  |  |
| --- | --- | --- |
| Selector + sel |  |  |
| Selector ~ sel |  |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |

<div class=intro> div.intro

<div> <p class=intro> div .intro // div > p.intro // div > p

**<section class=intro><div> .intro div //**

**Note:** An id /class name cannot start with a number!

|  |  |  |
| --- | --- | --- |
| **Selector** | **Example** | **Example description** |
| [*element.class*](https://www.w3schools.com/cssref/sel_element_class.asp) | p.intro | Selects only <p> elements with class="intro" |
| [*element,element,..*](https://www.w3schools.com/cssref/sel_element_comma.asp) | div, p | Selects all <div> elements and all <p> elements |

- use classes for styling. Don't use ids.

- use :first-child, :last-child and :nth-child to get at non-classed elements

- use ::before and ::after to create pseudo-elements for extra styling

- use multiple background images to apply as many as need any element

- use the Kellum Method for image replacement

- don't rely on the extra divs at the bottom. Use ::before and ::after

<https://www.w3schools.com/css/css_combinators.asp>

[class\*=”col-“]

.classA.classB

:first-child p:first-child Selects every <p> elements that is the first child of its parent

:last-child p:last-child Selects every <p> elements that is the last child of its parent

:nth-child(n) p:nth-child(2) Selects every <p> element that is the second child of its parent

:first-of-type p:first-of-type Selects every <p> element that is the first <p> element of its parent

:last-of-type p:last-of-type Selects every <p> element that is the last <p> element of its parent

:not(selector) :not(p) Selects every element that is not a <p> element

::after p::after Insert content after every <p> element

::before p::before Insert content before every <p> element

::first-letter p::first-letter Selects the first letter of every <p> element

::first-line p::first-line Selects the first line of every <p> element

::marker ::marker Selects the markers of list items

::selection p::selection Selects the portion of an element that is selected by a user

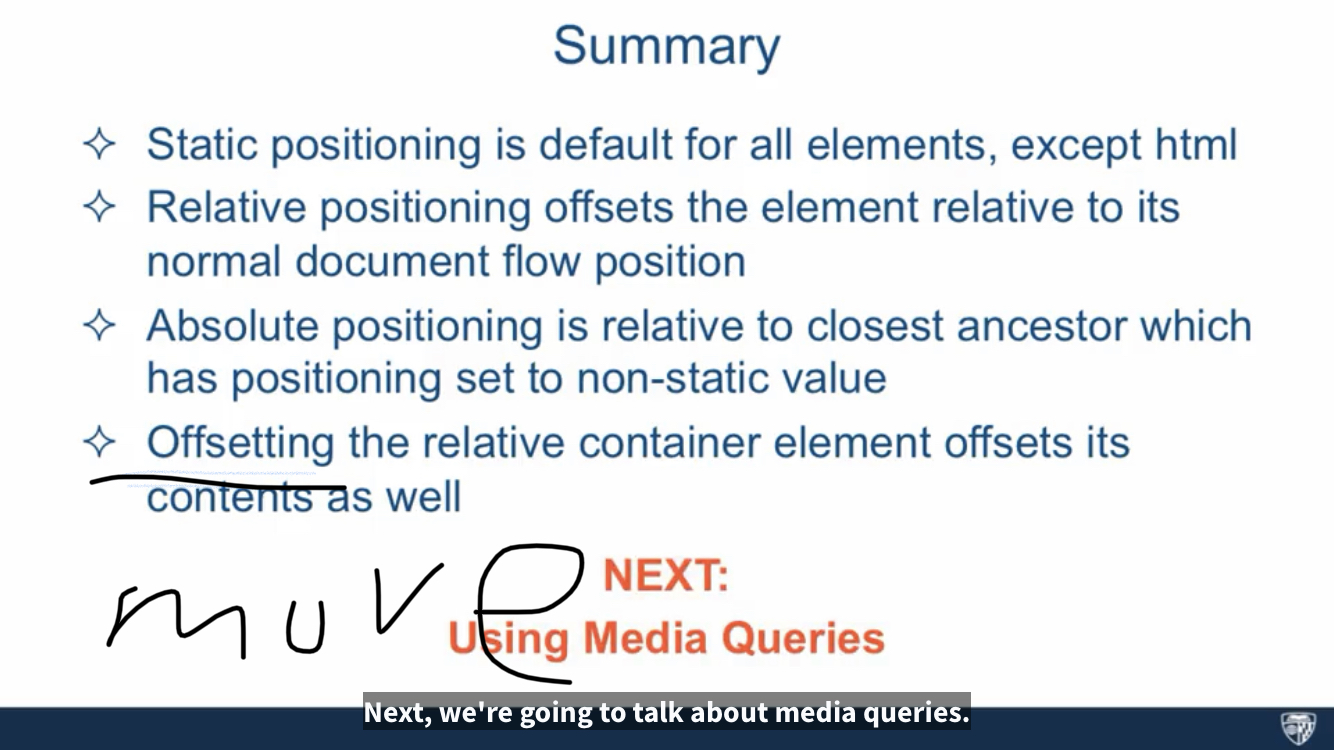
基本效果

|  |  |  |
| --- | --- | --- |
|  |  | 注 |
| background | body {   background: #ffffff url("img\_tree.png") no-repeat right top; } | * background-color * background-image * background-repeat * background-attachment * background-position |
| 位置  不會碌走attachment | background-repeat: repeat-x/y  background-position: right top;  background-attachment: scroll; scroll; fixed; local; |  |
| Clip剪 | background-clip: border-box (this is default): padding-box:  content-box  background-clip: text;-webkit-background-clip: text;color: transparent;  background-origin: padding-box|border-box|content-box|initial|inherit; | 字下面係圖既效果 |
| size | background-size: auto|*length*|cover|contain|initial|inherit;  auto= Default value  cover Resize the background image to cover the entire container, even if it has to stretch the image or cut a little bit off one of the edges |  |
| Repeat | background-repeat: repeat|repeat-x|repeat-y|no-repeat|initial|inherit; |  |
| border | p {   border-top-style: dotted;   border-right-style: solid;   border-bottom-style: dotted;   border-left-style: solid; } |  |
| margin | * auto - the browser calculates the margin * *%* - specifies a margin in % of the width of the containing element * inherit - specifies that the margin should be inherited from the parent element   左右：相加  上下：Margin Collapse    margin: auto; | **Tip:** Negative values are allowed.  You can set the margin property to auto to horizontally center the element within its container. |
| padding |  |  |
| Overflow  凸出黎  overflow-wrap | overflow-x: hidden; /\* Hide horizontal scrollbar \*/   overflow-y: scroll; /\* Add vertical scrollbar \*/  **Note:** The overflow property only works for block elements with a specified height.  **overflow-wrap: break-word:** |  |
| -position | Left center |  |

|  |  |  |
| --- | --- | --- |
| Box shadow | 右下blur 伸透  -webkit-box-shadow: 5px 5px 15px 5px #000000;  box-shadow: 5px 5px 15px 5px #000000; |  |
| Text-shadow |  |  |
| 透明 | opacity/transparency  **RGBA** | all of its child elements inherit |

|  |  |  |
| --- | --- | --- |
| Audio | <iframe src="audio/source.mp3" allow="autoplay" style="display:none" id="iframeAudio">  </iframe> |  |

position



* Float

margin左右加，上下冧

Float，taken out of normal doc flow，永不冧

Clear left：冇野可以在佢左邊float

flex

|  |  |  |
| --- | --- | --- |
| 外 |  |  |
| 內 | display: flex;  justify-content: center;  align-items: center; |  |

<https://www.casper.tw/css/2017/07/21/css-flex/>

|  |  |  |
| --- | --- | --- |
| Container | * display * flex-flow   + flex-direction 行/列   + flex-wrap 換行 * justify-content主軸對齊 * align-items交錯軸對齊   flex-wrap  超出範圍時是否換行的屬性，分為換行、不換行、換行時反轉。 | **display**: flex | inline-flex;  **flex-direction 行/列**: row | row-reverse | column | column-reverse;  **flex-wrap**: nowrap | wrap | wrap-reverse;  **flex-flow**: <'flex-direction'> || <'flex-wrap'>  **justify-content**: flex-start | flex-end | center | space-between | space-around;  **align-items**: flex-start | flex-end | center | baseline | stretch;  **align-content**: flex-start | flex-end | center | space-between | space-around | stretch; |
| Items | * flex   + flex-grow   + flex-shrink   + flex-basis * order   可以重新定義元件的排列順序，順序會依據數值的大小排列。   * align-self   可以調整內元件交錯軸的對齊設定(主軸線則不能另外做設定)，且可以個別設定單一元件的值。 |  |

flex 是縮寫，裡面依序包含三個屬性 flex-grow、flex-shrink 和 flex-basis，如果只設定一個則是 flex-grow。

* flex-grow: 元件的伸展性，是一個數值，當空間分配還有剩餘時的當前元件的伸展性，預設值為 0，如果設置為 0 則不會縮放。
* flex-shrink: 元件的收縮性: 元件的伸展性，是一個數值，當空間分配還不足時的當前元件的收縮性，預設值為 1，如果設置為 0 則不會縮放。
* flex-basis: 元件的基準值，可使用不同的單位值。

以下圖為例，flex-grow:為 2 的空間會佔更多。

Responsive

<meta name="viewport" content="width=device-width, initial-scale=1.0">

/\* Extra small devices (phones, 600px and down) \*/  
@media only screen and (max-width: 600px) {...}  
  
/\* Small devices (portrait tablets and large phones, 600px and up) \*/  
@media only screen and (min-width: 600px) {...}  
  
/\* Medium devices (landscape tablets, 768px and up) \*/  
@media only screen and (min-width: 768px) {...}  
  
/\* Large devices (laptops/desktops, 992px and up) \*/  
@media only screen and (min-width: 992px) {...}  
  
/\* Extra large devices (large laptops and desktops, 1200px and up) \*/  
@media only screen and (min-width: 1200px) {...}

* 如{中{，記得兩個 }}

|  |  |  |
| --- | --- | --- |
| 電腦 | <div class="col-lg-3 col-md-6"><p>...  <div class="row">  <div class="col-3 col-s-3">...</div>  <div class="col-6 col-s-9">...</div>  <div class="col-3 col-s-12">...</div>  </div>  For desktop:  The first and the third section will both span 3 columns each. The middle section will span 6 columns.  For tablets:  The first section will span 3 columns, the second will span 9, and the third section will be displayed below the first two sections, and it will span 12 columns: | /\* For desktop: \*/ .col-1 {width: 8.33%;} .col-2 {width: 16.66%;} .col-3 {width: 25%;} .col-4 {width: 33.33%;} .col-5 {width: 41.66%;} .col-6 {width: 50%;} .col-7 {width: 58.33%;} .col-8 {width: 66.66%;} .col-9 {width: 75%;} .col-10 {width: 83.33%;} .col-11 {width: 91.66%;} .col-12 {width: 100%;}  @media only screen and (max-width: 768px) {   /\* For mobile phones: \*/   [class\*="col-"] {     width: 100%;   } } |
| 手機先 | * 手機先 * 只變px 、 class name * Html 兩個class | /\* For mobile phones: \*/ [class\*="col-"] {   width: 100%; }  @media only screen and (min-width: 600px) {   /\* For tablets: \*/   .col-s-1 {width: 8.33%;}   .col-s-2 {width: 16.66%;}   .col-s-3 {width: 25%;}   .col-s-4 {width: 33.33%;}   .col-s-5 {width: 41.66%;}   .col-s-6 {width: 50%;}   .col-s-7 {width: 58.33%;}   .col-s-8 {width: 66.66%;}   .col-s-9 {width: 75%;}   .col-s-10 {width: 83.33%;}   .col-s-11 {width: 91.66%;}   .col-s-12 {width: 100%;} }  @media only screen and (min-width: 768px) {   /\* For desktop: \*/   .col-1 {width: 8.33%;}   .col-2 {width: 16.66%;}   .col-3 {width: 25%;}   .col-4 {width: 33.33%;}   .col-5 {width: 41.66%;}   .col-6 {width: 50%;}   .col-7 {width: 58.33%;}   .col-8 {width: 66.66%;}   .col-9 {width: 75%;}   .col-10 {width: 83.33%;}   .col-11 {width: 91.66%;}   .col-12 {width: 100%;} } |
|  |  |  |
|  |  |  |
| 橫向 |  | @media only screen and (orientation: landscape) {   body {     background-color: lightblue;   } } |
| hide |  | /\* If the screen size is 600px wide or less, hide the element \*/ @media only screen and (max-width: 600px) {   div.example {     display: none;   } } |
|  |  |  |

* width: 90%

margin-right,left :auto;

* Column 直行 、Row 橫列
* @media screen and (max-width:632px)

@media only screen and (max-width:632px)

Bootstrap

<https://getbootstrap.com/docs/5.3/getting-started/download/>

Body

<script src= ““js/jquery-…min.js” >….< /script>

<script src= “js/bootstrap.min.js”>….< /script>

<script src= “js/script.js>….< /script>

Grid

設計：要對齊 人眼

    <div class="container">

        <div class="row">

            <div class="col-md-4" ></div>

        </div>

    </div>

Container / container-fluid (有左右 15px padding)

col-size-span

size: xs / sm /md / lg (min 1200px in box) /

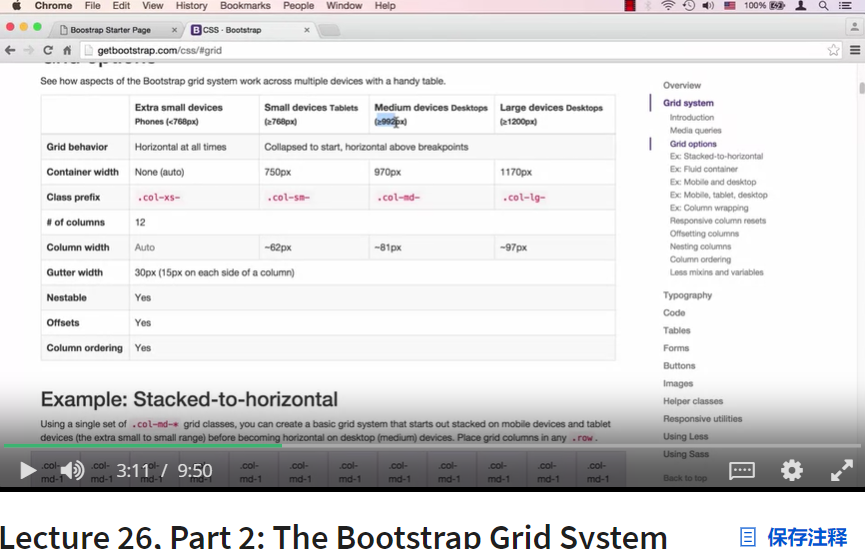
span: 1至12

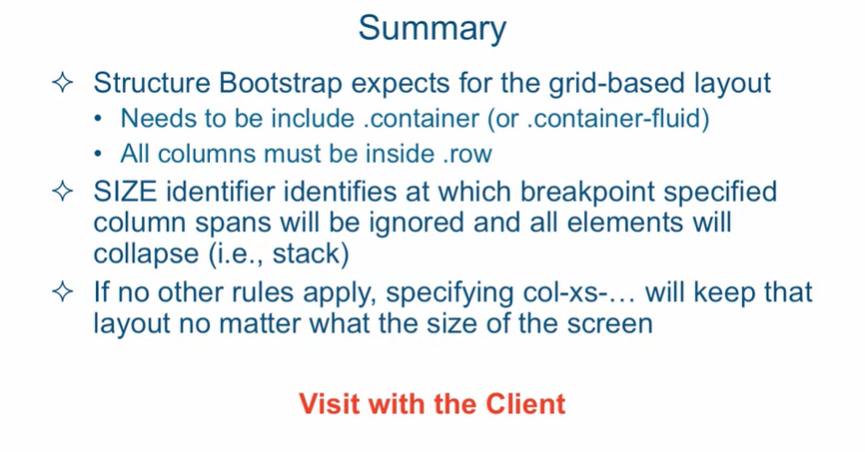
如有3格

class=”col-md-4 col-sm-6”

xs : never collapse

|  |
| --- |
| None、<768、大等768、大等992 |





# 真實整網 / 客人

餐廳

重點：

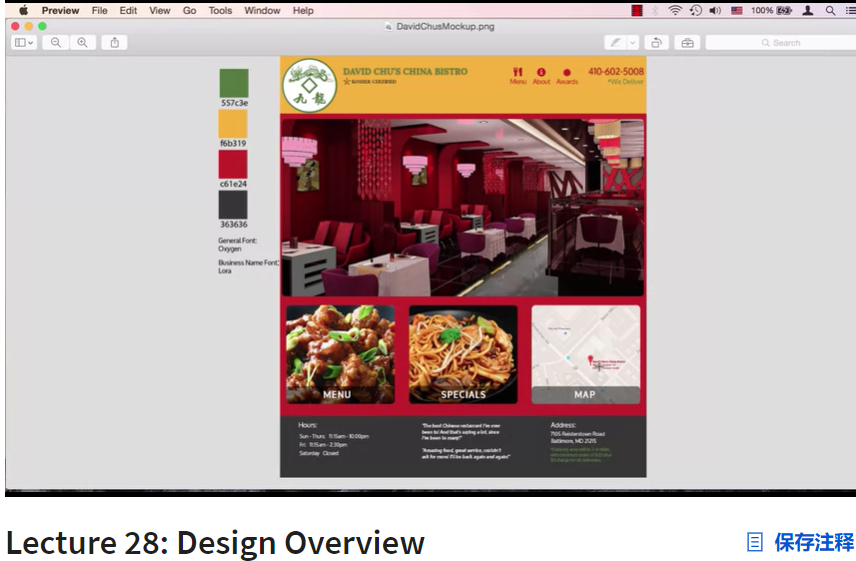
* Clients唔知想點，畀類似生意的reference佢地揀
* Less is more！叫客唔好放太多info
* Find a way for the Clients to invest in the project，如客畀產品photo錢$$$
* 搵一人專責resposible
* 一早講明 限定 FREE revision次數
* 查“web development client questionnaire”
* 如需要，搵人幫手 (EG designer、photographer)
* Get an idea 個客而家有乜，eg. Website、產品

第一步：

* 先Design layout、mockup，畀客 approve <https://balsamiq.com/>

1. 簡單mockup

2. design (PS、PPT、GIMP)，有details：圖、color主色調、font、logo font



第二步：

* 寫code
* Ground Rules and Overview of Setup

<https://fonts.google.com/>

# 例子

layout

重點：

* 位置：

1. position relative; / absolute [如一格格又position，會冧]

2.搵中間點 top 50%

left 50%

transform: translate(-50%, -50%);

-webkit-transform: translate(-50%, -50%);

* Float 要加 clear both
* 圖Responsive：max-width: 80% // 平分3格：width 33.3% ，4格 25%
* 一格格：外div flex，內div

justify-content: center;

align-items: center;

|  |  |  |
| --- | --- | --- |
|  | html | Css |
| 圖上字 | <div class=”box”>  <div class=”content”>  <p>文字</p>  </div>  </div> | .b{  background-image: url();  background-size: 100% 100%;  background-repeat: no-repeat;  width:  height:  min-height:  position: relative;  .c{  Position: absolute;  Color:  Top, left,  Body{  Margin:0; |
| 圖上字  加框、filter | <div class=”box”>  <div class=”content”>  <p>文字</p>  </div>  </div> | .b{  background-image: url(background.jpg);  background-size: 100% 100%;  background-repeat: no-repeat;  min-height :768px;  clear: both;  position: relative;  .c{  backdrop-filter:saturate(300%);  background-color: rgba(255, 255, 255, 0.3);  width: 500px;  height: 500px;  border-radius: 30px;  color: blanchedalmond;  text-align: center;  position: absolute;  top: 150px;  left: 300px; |
| 字+背色 | Same  <div class="email">  <h4>Subscribe To </h4>  <p>Subscribe for ….</p>  <form action="form.submit.php" method="post">  <input type="email" placeholder="彩蛋" name="email" value="email"/>  <input type="submit" value="submit">  </form></div> | .email{  background-color: cornsilk;  width: 100%;  height: 200px;  text-align: center;  padding-top: 50px;  .c{  不用 |
| 色中  打橫3格 |  | B{  Back color  Width 100%  Height: px  c{  Display:flex  Float: left  Width 100%  Photo2{  Width 33.3% (平分3格)  text-align: center; |
| 色中  色中  字+圖 |  | max-width: 80%; 有responsive 效果 |

Link, header, footer

|  |  |  |
| --- | --- | --- |
| Menu  右邊 |  | Nav ul{  Float: right;  List-style-type:none;  Nav li{  Float: left;  Display:inline;  Nav a:link, nav a:visited{  Text-dec:none;  Color:  Nav a:hover{  Color:  header{      position: sticky;      top:0; |
| 鼠移上 | <div class="welcome">  <ul>  <li><ahref="index.html">首頁</a></li>  </ul>  </div> | .b {  Pos: relat  .b :hover ::before{  Pos: absol  Back-color:  Color:  Content:”字呀”;  Font-size |
|  |  |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

# ============

# JavaScript

|  |  |
| --- | --- |
| Variables |  |
| Function |  |
| Return |  |
|  |  |

Variables

What characters are valid for JavaScript variable names?

Rules

* A letter, dollar sign($), or underscore (\_) must make up the first character. A number cannot be the initial character.
* Any letter, number, or underscore can complete the variable name. You cannot use some characters, such as spaces, symbols, and punctuation.
* Names of variables are case-sensitive.
* The name of the variable can be of any limit as per your need.
* The name of a variable cannot contain a reserved word in JavaScript.

Function

<https://www.youtube.com/watch?v=yZwlW5INhgk&ab_channel=GrandmaCan-%E6%88%91%E9%98%BF%E5%AC%A4%E9%83%BD%E6%9C%83>

1:05:30

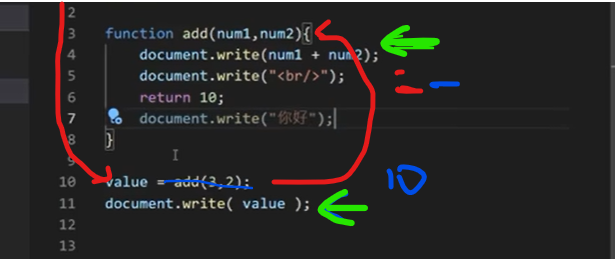
當係Subway叫麵包食, sayHello係個熱狗麵包, ()裡面就係材料, 放咩材料入去都得，而sayHello熱狗麵包放落焗爐出黎就變 Diu Nei,材料 ， 所以 sayHello('HK') 出黎就係 Diu Nei, HK

Return：

1. 回傳的值，會覆蓋掉原先的call (呼叫)，方便後來再作運算、處理

2. Function碰到return，就會出跳來，跳出function，之後下面再寫咩都不會run

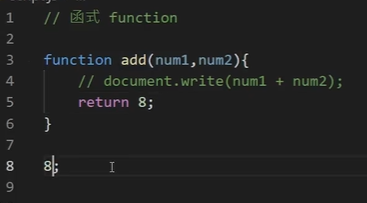
|  |  |
| --- | --- |
| 因為 | 8 |
|  | 9，不會出「你好」 |
|  | 5  10 |



碰到10 call，才執行3-7 function，碰到6 return，覆蓋了 add(3,2) ，value = 10

|  |  |
| --- | --- |
|  |  |

意思是這樣：



If / else if / else

• == 等於

• != 不等於

• > 大過

• < 細過

• >= 大過或等於

• <= 細過或等於

* && 且，兩樣都要成立
* || 或，其中一樣成立

|  |  |
| --- | --- |
|  |  |
|  | Var score = 90既話  你給我100 |
|  | Var score = 90既話  我給你1000 |
|  | 你給我100 |
| 要3個No. 回傳最大值 |  |

Object

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

While

<https://www.youtube.com/watch?v=yZwlW5INhgk&ab_channel=GrandmaCan-%E6%88%91%E9%98%BF%E5%AC%A4%E9%83%BD%E6%9C%83>

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |

# ============

|  |  |
| --- | --- |
|  |  |

# Python | AI machine learning / SEO google / 談判 / UI UX / 設計

<https://www.coursera.org/professional-certificates/google-digital-marketing-ecommerce>

<https://www.coursera.org/learn/negotiation-skills>

<https://www.coursera.org/learn/negotiation>

<https://www.coursera.org/search?query=free%20courses&=null&utm_source=gg&utm_medium=sem&utm_content=Sitelink-freecourses-searchquery&campaignid=19731260869&adgroupid=141562115450&device=c&keyword=coursera%20fees&matchtype=b&network=g&devicemodel=&adpostion=&creativeid=649154929096&hide_mobile_promo=null&gclid=CjwKCAjwsvujBhAXEiwA_UXnAAVpx7HEJHnD8nu0-Ue6zAUk2GxFECJBSgsDFmpHO3QIUN5x8qvk2xoCDqEQAvD_BwE&index=prod_all_launched_products_term_optimization>

<https://www.coursera.org/learn/brand>

<https://www.coursera.org/learn/logic-introduction>

# 心得

堅持不懈

事業上的成功並不容易。幸運的是，這也不太難。考慮一下學習編碼成為你日常生活一部分的新嘗試。

盡可能讓它成為一種例行公事。希望它能像這樣工作：

你醒來，

你刷牙，

你去辦一些事，

然後你編寫並學習編碼。

顯然，有些事情比如你的學校義務、你的日常工作、或者你需要去的其他地方以及你需要做的事情。

但是，如果您不定期編碼（最好是每天），那麼您的進度將會變慢。嘗試每天留出一些時間來持續編碼和學習。堅持是關鍵。

立即開始構建簡單的應用程序

不要等到“學得足夠多”。總是有更多東西需要學習，最好現在就開始任何類型的簡單項目。

即使只是從這個專業領域中獲取代碼並進行修改，也會對您的信心和獲取新知識的速度產生奇蹟。

而且，練習得越多，你就能更好地記住所學的內容。

擁有可以向他人展示的自己的項目 - 無論多麼小或簡單，都顯示出良好的記錄和奉獻精神。這可能會給你未來的雇主留下深刻的印象，所以從今天開始吧。

設置 GitHub 帳戶

既然我們談論的是個人項目，那麼請前往

GitHub

並立即設置您的開發者資料。擁有一個帳戶非常重要，因為您可以將所有項目保存在一個可以從任何計算機訪問的位置。

您幾乎可以將 GitHub 帳戶視為額外的腦力。無論多久以前，你所做的一切都會留在那裡，等待你去窺視並重新熟悉。

結對計劃

嘗試找到與您水平相當或知識水平比您稍高的人，並要求他們建立定期的結對編程課程。

這很有效，因為結對編程夥伴可以加快你的學習速度。你也有一個要對之負責的人。

開始寫一個編碼博客

技術溝通對於開發人員來說很重要，就像其他事情一樣，通過練習你會變得更好。

創建編碼博客與擁有 GitHub 帳戶的方式相同，並且還有一些額外的好處：

它顯示出更多的奉獻精神 - 這會增加您被雇用的機會

它可以幫助您嘗試不同的技術

建立自己的網站本身就是一種實用的學習，也是一個可以添加到您的簡歷中的項目

就開源項目進行協作

即使您剛剛起步並且正在努力進入這個領域，您仍然可以成為開源項目的寶貴貢獻者。

有如此多的開源項目需要各種貢獻者。

即使通過修復文檔文件中的一些拼寫錯誤來為項目做出貢獻，也是更多地參與並讓自己投入其中的一個很好的開始。

獲得證書

獲得認證總是一件好事。您現在正在閱讀本課程，這一事實證實您即將獲得 Coursera 頒發的結業證書！

保持積極的態度

與任何值得做的事情一樣，您有時可能會感到疲倦，不明白某件事是如何運作的，甚至可能想放棄。

記住要保持一致。

生活中總會有起起落落，但有時想想你迄今為止所取得的所有成就並以此作為堅持下去的動力是值得的。

從未停止學習

IT 領域總是有更多東西需要學習，這可能是它最好的一點。這正是它的樂趣所在，並為每個開發人員提供了在職業生涯中取得進步的機會。

# 如何發現工作機會

學習如何使用 JavaScript 編程可以幫助您為各種工作機會做好準備。部分原因是它擴展了開發人員構建內容的可能性。

JavaScript 是最受歡迎的編程語言之一，因為幾乎所有活躍的網站都使用它。它的多功能性使開發人員能夠使用流行的庫、插件和框架（例如 React），從而提高效率和生產力。

由於 JavaScript 編程帶來的靈活性，您可能想要從事不同的職業也就不足為奇了。然而，無論你選擇什麼職業道路，你總是想學習其他技術，如 HTML、CSS、React、Node.js 或 Python，這樣你就更有市場。

讓我們介紹一下如果您知道如何使用 JavaScript 編程，您可以獲得的一些最常見的角色。

**移動開發人員** 隨著移動設備訪問互聯網的使用不斷增加，對移動開發人員的需求不斷增加。移動開發人員專門為 Google 的 Android 和 Apple 的 iOS 等平台構建應用程序。許多開發人員選擇使用 React Native，這使他們能夠使用 JavaScript 構建一個可在 Android 和 iOS 設備上運行的應用程序。移動開發人員與 UX 和 UI 設計師合作，並使用 React 的 UI 功能來實現客戶將使用的功能。移動開發人員還確保應用程序的前端和後端無縫工作。移動開發人員可能擁有的其他技能包括 HTML、CSS、Java、Kotlin、Objective-C、C++、C# 等。

**前端開發人員** 顧名思義，前端開發人員構建網站和應用程序中面向用戶的部分。他們與設計師密切合作，通過使用 HTML、CSS，當然還有 JavaScript 進行編碼來實現視覺和交互元素。他們還使用 React 等庫和框架來節省時間並提高工作效率。前端開發人員還可能負責確保最終用戶擁有良好的用戶體驗，並且網站和應用程序的行為符合預期並且沒有錯誤和錯誤。

**後端開發人員** 後端開發人員負責網站和應用程序的後端工作。他們可以使用 JavaScript 和 Node.js 來開發後端功能。這些功能包括流媒體和基於聊天的應用程序，以及 JSON API 和無服務器功能。後端開發人員擁有額外的技能，包括Python、API、雲基礎設施和數據庫的專業工作知識。

**全棧開發人員** 正如您可能猜到的那樣，全棧開發人員同時負責構建網站和應用程序的前端和後端。因此，這些專業人員結合了這兩個領域的技能。他們可以使用 React 等框架來處理前端，使用 Node.js 來處理後端。他們還應用 JavaScript 之外的技能來構建網站和應用程序。

**您未來的職業旅程**

當您作為一名專業開發人員開始令人興奮的職業生涯時，您將意識到您將逐步擴展您的技能，以涵蓋除 JavaScript 之外的廣泛技術和編程語言。

如果您想了解這些領域的機會，您可能需要檢查您最喜歡的求職網站或應用程序，並尋找與 JavaScript 相關的工作。你會發現並不缺少機會。當您開始閱讀有關這些職位的更多信息時，您會發現 JavaScript 通常只是雇主正在尋找的能力之一。

但不用擔心，隨著您在學習和職業生涯中的進步，您將進一步發展您的技能，以專注於您想要遵循的職業道路。