

MARIAN ONLINE



ห้ามก็อปเนื้อหา และนำสไลด์ไปเผยแพร่ ฝ่าผิดปรบ ละเมิดความดำเนินคดี



ເວສາດກາສອງ

HERO WEBSITE / PROTFOLIO



Web Application គេ ?



Web Application คือ ?

Web Application คือแอปที่ถูกเขียนขึ้นมาให้สามารถเปิดใช้ใน Web browser ได้โดยตรง ไม่ต้องโหลด Application แบบเต็มๆ ลงเครื่อง ทำให้โดยรวมแล้วกินทรัพยากรค่อนข้างต่ำ สามารถเปิดใช้งานได้ไว



Website and Web Application ?





Website and Web Application ?



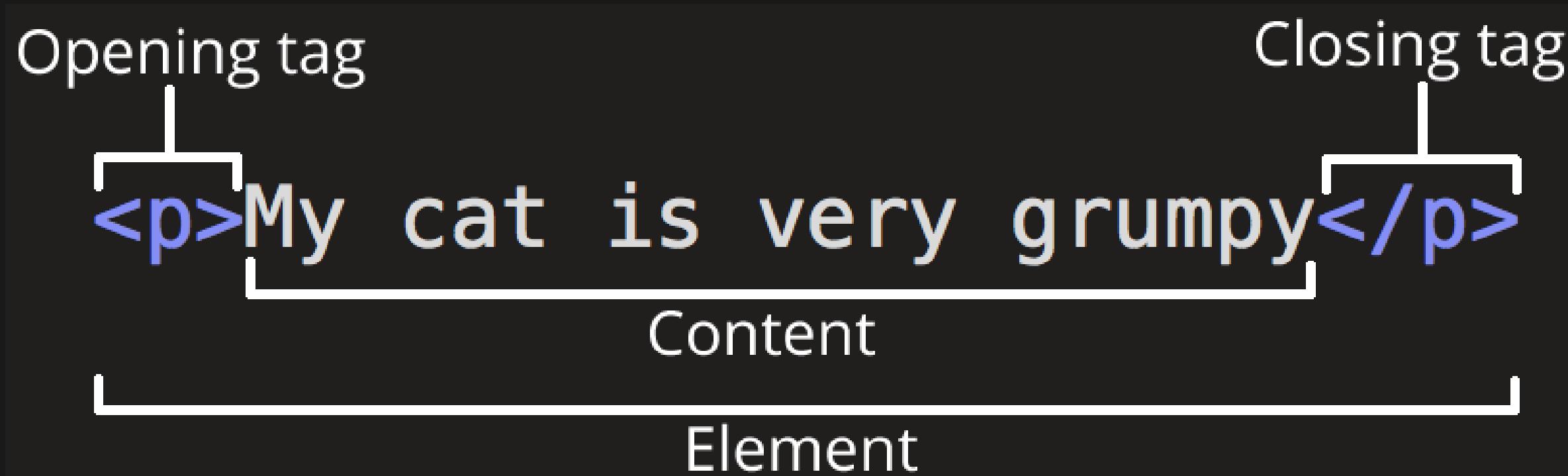
มาเริ่มทำความรู้จัก HTML กัน

HYPER TEXT MARKUP LANGUAGE (HTML) คือ โครงสร้างของเว็บไซต์ทั้งหมด
(เหมือนบ้านก็มีแต่โครงสร้าง และเสา แต่ยังไม่ได้ตกแต่ง)





Anatomy of an HTML element





Attributes

Attribute

```
<p class="editor-note">My cat is very grumpy</p>
```



Anatomy of an HTML document

```
<!DOCTYPE html>
<html lang="en-US">
  <head>
    <meta charset="utf-8" />
    <title>My test page</title>
  </head>
  <body>
    <p>This is my page</p>
  </body>
</html>
```



Folder Structure





What is the HTML head?

```
<head>
  <meta charset="utf-8" />
  <title>My test page</title>
</head>
```



Adding a title

```
<title>My testpage</title>
```



Metadata: the <meta> element



<meta charset="utf-8" />



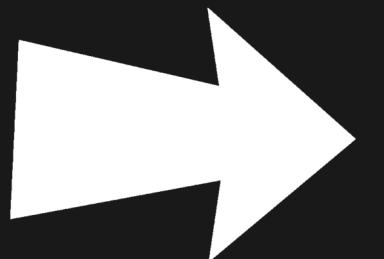
เนื้อหาที่เราจะเลสลงบนหน้าเว็บ



<h1>My Heading</h1> = ใช้เน้นหัวข้อความ

```
<h1>Header 1</h1>
<h2>Header 2</h2>
<h3>Header 3</h3>
<h4>Header 4</h4>
<h5>Header 5</h5>
```

OUTPUT



```
Header 1
Header 2
Header 3
Header 4
Header 5
```



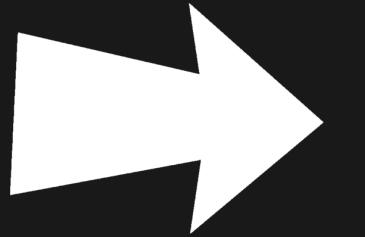
เนื้อหาที่เราจะlessonหน้าเว็บ



< p > Paragraph </ p > = ใช้จัดพารากราฟ
< i > Italic </ i > = ใช้กำหนดให้ตัวอักษรเป็นตัวเอียง
< b > Bold text </ b > = ใช้กำหนดให้ตัวอักษรเป็นตัวหนา
< u > Underline </ u > = ใช้สร้างเส้นใต้ตัวอักษร

```
<p>
Hello
<b>World</b>
<i>Marian</i></i>
<u>Online</u>
</p>
```

OUTPUT



Hello **World** *Marian* Online

เนื้อหาที่เราจะlessonหน้าเว็บ

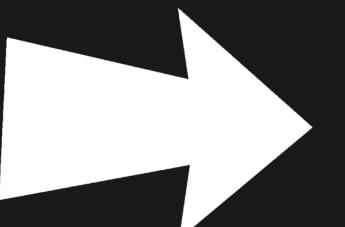


<hr> = ใช้สร้างเส้นคั่น

 = ใช้สำหรับการขึ้นบรรทัดใหม่

```
<p>  
สวัสดี  
<br><b>ชาวโลก</b>  
<hr><i>ทุกคน</i>  
<br><n>ทำไรอยู่</n>  
</p>
```

OUTPUT



สวัสดี
ชาวโลก

ทุกคน
ทำไรกันอยู่



เนื้อหาที่เราจะสอนบนหน้าเว็บ



ใช้เชื่อมต่อกับอีกเว็บ

< a href = "url / ชื่อ ID "> Link </ a >

OUTPUT
→

Link

แสดงรูป

< img src = "url / ชื่อไฟล์" width="ขนาด" height="ขนาด" >

OUTPUT
→



ถ้าจะเอาวีดีโอของยูทูปเข้ามาใช้

- ↔ วนซ้ำ
- ↪ คัดลอก URL ของวีดีโอ
- ↪ คัดลอก URL วีดีโอที่เวลาปัจจุบัน
- <> คัดลอกโค้ดสำหรับฝัง

แสดงวีดีโอ

< video width="ขนาด" height="ขนาด" >
< source src="url / ชื่อไฟล์" >
</ video >



เนื้อหาที่เราจะเลสลงบนหน้าเว็บ

Bootstrap5



Bootstrap 5



TemplateToaster
Website Builder | Since 2009



เนื้อหาที่เราจะเลสลงบนหน้าเว็บ

Font Awesome





HOMEWORK

สร้าง Website E-Commerce
Use ONLY HTML!

Requirement

- Bootstrap5
- Font Awesome
- Video
- Image
- 3 page++



HTML INPUT TYPES CHEATSHEET

```
1 <input type="text">
```

this is a text field

Type: text

INPUT TYPE: TEXT

```
1 <input type="number">
```

7

Type: number

INPUT TYPE: NUMBER

```
1 <input type="email">
```

username@gmail.com

Type: email

INPUT TYPE: EMAIL

```
1 <input type="password">
```

Type: password

INPUT TYPE: PASSWORD

```
1 <input type="datetime-local">
```

dd-mm-yyyy --::--

March, 2022

Mo	Tu	We	Th	Fr	Sa	Su
28	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31	1	2	3
4	5	6	7	8	9	10

Clear Today

INPUT TYPE: DATETIME-LOCAL

```
1 <input type="time">
```

--::--

01	08	PM
02	09	AM
03	10	
04	11	
05	12	
06	13	
07	14	

INPUT TYPE: TIME



The image shows six examples of different input types in a code editor interface. Each example consists of a code snippet on the left and a corresponding browser input field on the right.

- INPUT TYPE: URL**: The code is `<input type="url">`. The browser shows a text input field with the placeholder "Type: url" and a tooltip containing the URL "twitter.com/Aakash_codes".
- INPUT TYPE: SEARCH**: The code is `<input type="search">`. The browser shows a text input field with the placeholder "Type: search" and a tooltip containing the text "Web Developer".
- INPUT TYPE: DATE**: The code is `<input type="date">`. The browser shows a date picker with the date "March, 2022" selected. The tooltip shows the date "dd-mm-yyyy".
- INPUT TYPE: WEEK**: The code is `<input type="week">`. The browser shows a week picker with the week "Week 12, 2022" selected. The tooltip shows the date "Week ---".
- INPUT TYPE: RESET**: The code is `<input type="reset">`. The browser shows a button labeled "Reset". The tooltip shows the text "Type: reset".
- INPUT TYPE: RANGE**: The code is `<input type="range">`. The browser shows a horizontal slider with a blue track and a white slider handle. The tooltip shows the text "Type: range".



The image shows five separate browser windows demonstrating different input types:

- INPUT TYPE: COLOR**: Shows an input field with a color picker. The color is set to approximately #E91E63. The code is <input type="color">.
- INPUT TYPE: SUBMIT**: Shows an input field with the value "Submit". The code is <input type="submit">.
- INPUT TYPE: CHECKBOX**: Shows an input field with the checked attribute. The code is <input type="checkbox">.
- INPUT TYPE: RADIO**: Shows an input field with the checked attribute. The code is <input type="radio">.
- INPUT TYPE: FILE**: Shows an input field with the value "Choose File No file chosen". The code is <input type="file">.



มาเริ่มทำความรู้จัก CSS กัน



CASCADING STYLE SHEETS (CSS) คือ ช่วยในการจัดรูปแบบการแสดงผลของ HTML
(เหมือนบ้านที่ถูกตกแต่ง)





All Css Units Cheatsheet

- ☀️ **px** : Absolute Length
- ☀️ **rem** : Relative to the `font-size` of the root Element
- ☀️ **em** : Relative to the `font-size` of the Element
- ☀️ **%** : Relative to the parent Element
- ☀️ **vw** : Relative to the viewport's width, $1vw = 1\% * \text{viewport's width}$
- ☀️ **vh** : Relative to the viewport's height, $1vh = 1\% * \text{viewport's height}$
- ☀️ **vmin** : Relative to the viewport's smaller dimension, $1vmin = \min(1vh, 1vw)$
- ☀️ **vmax** : Relative to the viewport's larger dimension, $1vmax = \max(1vh, 1vw)$
- ☀️ **ch** : Relative to the width of the glyph "0" of the element's font
- ☀️ **in** : Inches $1in = 2.54cm = 96px$
- ☀️ **pc** : Picas $1pc = 1in / 6 = 16px$
- ☀️ **pt** : Points $1pt = 1in / 72 = 1.333px$ (approximately)
- ☀️ **cm** : Centimeters $1cm = 1in / 2.54 = 37.8px$ (approximately)
- ☀️ **mm** : Millimeters $1mm = 1cm / 10 = 3.78px$ (approximately)





All Css Selector types Explained

✳️ * : Universal Selector

- Selects all elements of any type on HTML document.

```
* {  
    color: blue;  
}
```

This rule will change every HTML element on the page to have blue text.

✳️ type : type Selector

- Selects elements by node name, selects all elements of the given type within a given document. Ex: **div, span**

```
/* All <a> elements. */  
a {  
    color: red;  
}
```

This rule will change every anchor element on the page to have red color text.

✳️ .class-name : Class Selector

- Selects elements based on the content of **class="xxx"** attribute

```
/* All elements with class="spacious" */  
.spacious {  
    margin: 2em;  
}
```

This rule will change add margin of 2em to every element with class attribute of spacious

✳️ #id-name : Id Selector

- Selects an element based on the value of its **id="xxx"** attribute.

```
/* The element with id="demo" */  
.demo {  
    padding: 2em;  
}
```

This rule will change add padding of 2em to the element with id attribute of demo



[attr] : Attribute Selector

- matches elements based on the presence or value of a **given attribute**.

```
/* Select <a> element with a title attribute */  
a [title] {  
    color: blue;  
}
```

This rule will change the color of **<a>** element with title attribute to blue.



Grouping Selectors

- Grouping selector **(,)** can select multiple selector at once.

```
/* Selects the div & span and change their color to red */  
div,  
span {  
    color: red;  
}
```

This rule will change the color of **<div>** & **** element to red.



Compound Selectors

- We can combine selectors to increase specificity & readability.

```
/* Only select the <a> element which also has a class of spacious. */  
a.spacious {  
    color: red;  
}
```

This rule will change the color of **only a** element which also has a class of spacious.

All Css Combinators:-



Descendant Combinator

- A Descendant Combinator allows us to target a child element.

Select all child & grandchilids of a parent.



```
/* Selects all <strong> elements that are child of p */  
p strong {  
    color: red;  
}
```



This rule will change the color of **elements that are child of

to blue.**



Child Combinator

- Child Combinator helps us to select direct child of the parent.

Selects only direct children & not the grandchildren of a parent.



```
/* Selects all <strong> elements that are direct child of p */  
p > strong {  
    color: blue;  
}
```



This rule will change the color of **elements that are direct child of

to blue.**



General sibling Combinator

- A general sibling works in a way that even if the sibling before the child doesn't the same it will still select it.



```
/* Selects <strong> elements even if there is something different before last <strong> of <p> it will still select it */  
p ~ strong {  
    color: red;  
}
```



This rule will change all the **elements even if the element before the child is not same.**



Adjacent Sibling Combinator

- Adjacent sibling works in a way that it will only select a child matches the first element before it.



```
/* Selects <strong> elements only if it matches the first element before it */  
p + strong {  
    color: blue;  
}
```



This rule will change the color of **elements only if matches the first element before it of**





Pseudo Classes & Elements Selector:-



: Pseudo Classes

- The `:` pseudo allow the selection of elements based on **state information** that not contained in the **document tree**.

Ex: `a:visited`



```
/* Any button over which the user's
   pointer is hovering will change
   its background to blue */
button:hover {
  color: blue;
}
```



This rule will change the color of `<button>` element to blue when user pointer is hovered



:: Pseudo Elements

- The `::` pseudo represent entities that are not included in HTML

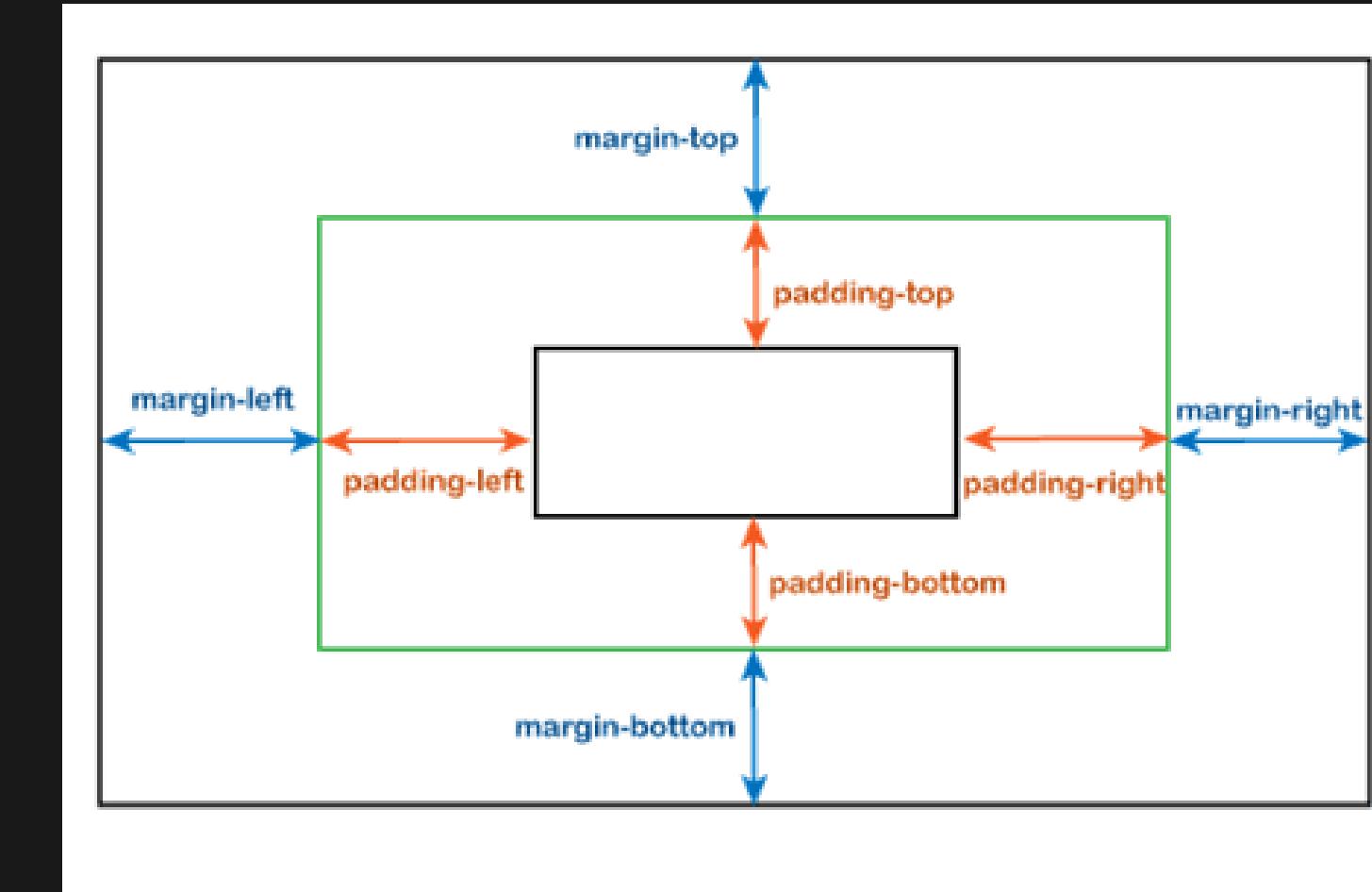
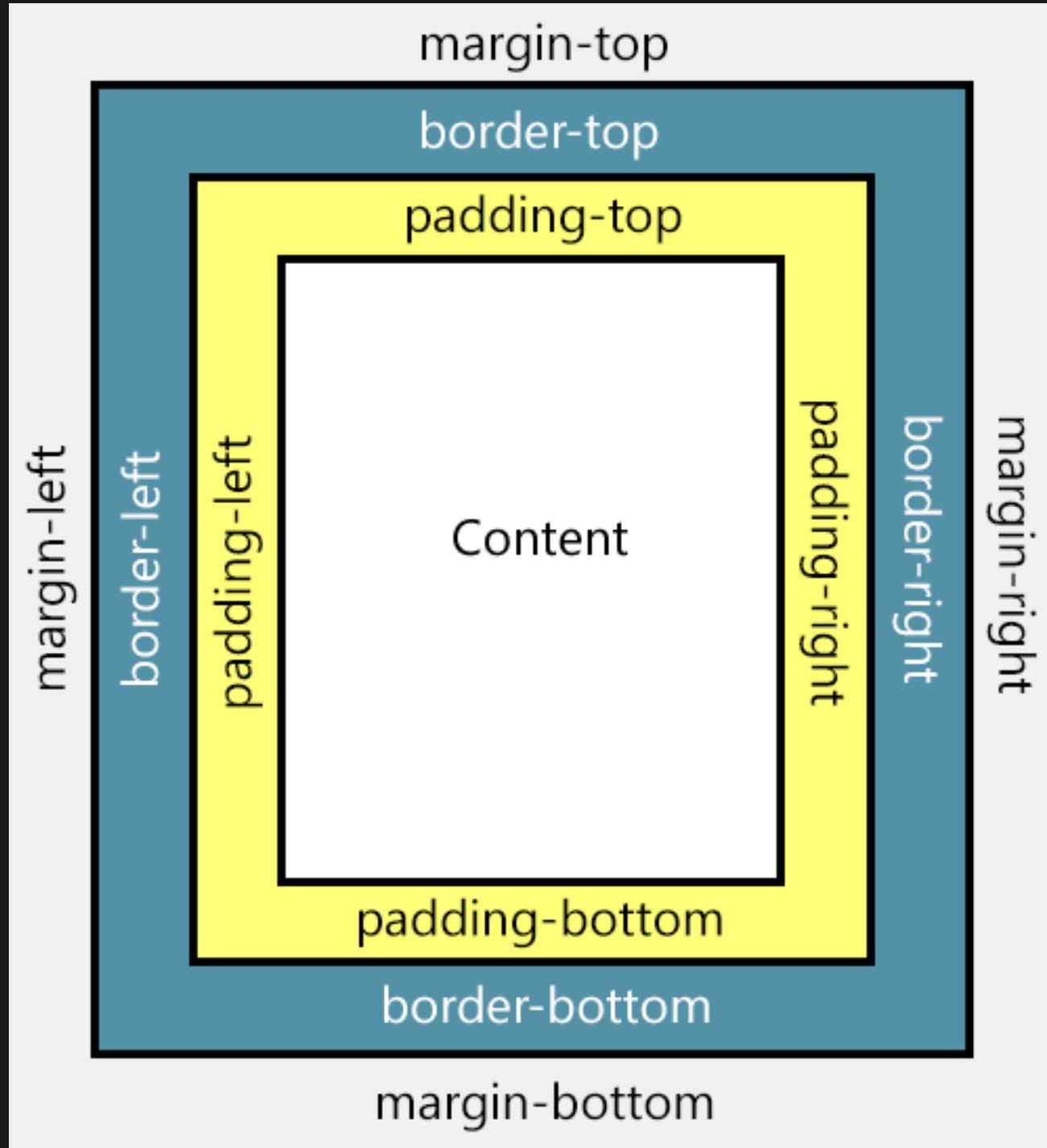
Ex: `p::first-line`



```
/* The first line of every <p> element. */
p::first-line {
  color: blue;
}
```



This rule will change the color of every first line of p element to blue



Border = เส้นขอบ

Padding = กำหนดระยะห่างในเส้นขอบ

Margin = กำหนดระยะห่างนอกเส้นขอบ



Css Flexbox Ultimate CheatSheet

CONTAINER {PARENT} PROPERTIES

DISPLAY

Enables Flex For All Children



`display: flex`



`display: inline-flex`

JUSTIFY-CONTENT

Attempts to distribute extra space on main axis



`flex-start`



`flex-end`



`center`



`space-between`



`space-around`

ALIGN-ITEMS

Determine how items are laid out on the cross-axis.



`flex-start`



`flex-end`



`center`



`baseline`



`stretch`

ALIGN-CONTENT

Only has an effect with more than one line of content.



`flex-start`



`flex-end`



`center`



`space-between`



`space-around`



`stretch`

FLEX-DIRECTION

Establishes the main axis.



`flex-direction: row`



`flex-direction: row-reverse`



`flex-direction: column`



`flex-direction: column-reverse`

FLEX-WRAP

Wraps items if they can't be made to fit on one line.



`flex-wrap: no-wrap`



`flex-wrap: wrap`



`flex-wrap: wrap-reverse`





ITEM {CHILDREN} PROPERTIES

FLEX-GROW

Allows you to determine how each child is allowed to grow as a part of a whole.



`flex-grow: 1;`
(Applied to all items)



`flex-grow: (1, 2 & 3)`

FLEX-BASIS

Define the size of an element before remaining space is distributed.



`first item 20%;`
`second item 40%`

FLEX-SHRINK

Allows an item to shrink if necessary. Only really useful with a set size or flex-basis.



`both want to be 100%`
wide, 2nd item
has `flex-shrink: 2`

ALIGN-SELF

Sets alignment for individual item.



`3rd item has`
`align-self: flex-end`

ORDER

The order property specifies the order of the flex items.



`order: -1;` on 3rd item

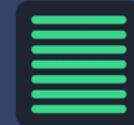


CSS Grid Ultimate CheatSheet

CONTAINER {PARENT} PROPERTIES

DISPLAY

HTML element becomes a grid container.



`display: grid;`



`display: inline-grid;`

GRID-TEMPLATE

Defines the rows & columns of the grid.



`grid-template-columns: 14px 14px 14px;
grid-template-rows: 14px 14px 14px;`



`grid-template-columns: repeat(3, 12px);
grid-template-rows: repeat(3, 14px)`



`grid-template-columns: 5px auto 5px;
grid-template-rows: 5px auto 5px;`



`grid-template-columns: 10% 10% auto;
grid-template-rows: 10% 10% auto;`

GRID-GAP

Defines the size of column & row gutters



`grid-gap: 14px;`



`grid-gap: 1px 14px;`



`grid-row-gap: 1px;
grid-column-gap: 14px ;`

Note: You can also use gap property which is very similar go grid-gap & works for both flexbox & grid.





ITEM {CHILDREN} PROPERTIES

GRID-ROW

Determines an items row-based location within the grid.



`grid-row-start: 1;
grid-row-end: 3;`



`grid-row-start: span 3`



`grid-row-start: 2;
grid-row-end: 4;`



`grid-row: 1 / 3;`



`grid-row: 1 / span 3;`

GRID-COLUMN

Determines an items column-based location within the grid.



`grid-column-start: 1;
grid-column-end: 3;`



`grid-column-start: span 3;`



`grid-column-start: 2;
grid-column-end: 4;`



`grid-column: 2 / 3;`



`grid-column: 2 / span 2;`

GRID-ROW + GRID-COLUMN

Combining grid rows with grid columns.



`grid-row: 1 / span 2;
grid-column: 1 / span 2;`



`grid-row: 2 / span 2;
grid-column: 2 / span 2;`

JUSTIFY-SELF

Aligns content for a specific grid item along the row axis.



`justify-self: start;`



`justify-self: end;`



`justify-self: center;`



`justify-self: stretch;`

ALIGN-SELF

Aligns content for a specific grid item along the column axis.



`align-self: start;`



`align-self: end;`



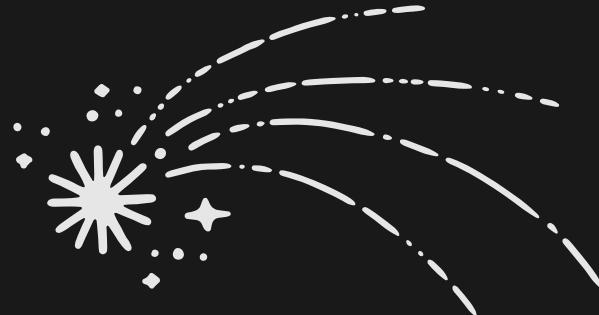
`align-self: center;`



`align-self: stretch;`



HOMEWORK



Instagram

- Home
- Search
- Explore
- Reels
- Messages
- Notifications
- Create
- Profile

The collage includes:

- A woman in a black top at a bar.
- A woman in a floral bikini.
- A hand holding a digital timer showing 999977.
- A woman in a white outfit taking a mirror selfie.
- A post from @true.loves (@true_loves2543) with Thai text: "ไม่ใช่แฟนกูจะไม่สนใจเลยลักษัน" and "กิมมิ่งแพนกุ".
- Two women and a man standing together with birth years: Me: 2001, Twin: 2002, Bro: 2006.
- A woman in a white bikini.
- A chessboard diagram titled "MATE IN 9".
- A woman in a pink bikini on a beach.
- A woman in a white top.

More





Git Installation

ไปที่ <https://git-scm.com/>
โหลดโปรแกรม Git (ถ้ายังไม่ได้ติดตั้ง)

 **git** --distributed-is-the-new-centralized

Git is a [free and open source](#) distributed version control system designed to handle everything from small to very large projects with speed and efficiency.

Git is [easy to learn](#) and has a [tiny footprint with lightning fast performance](#). It outclasses SCM tools like Subversion, CVS, Perforce, and ClearCase with features like [cheap local branching](#), [convenient staging areas](#), and [multiple workflows](#).



 **About**
The advantages of Git compared to other source control systems.

 **Downloads**
GUI clients and binary releases for all major platforms.

 **Documentation**
Command reference pages, Pro Git book content, videos and other material.

 **Community**
Get involved! Bug reporting, mailing list, chat, development and more.

Latest source Release
2.40.0
[Release Notes \(2023-03-12\)](#)

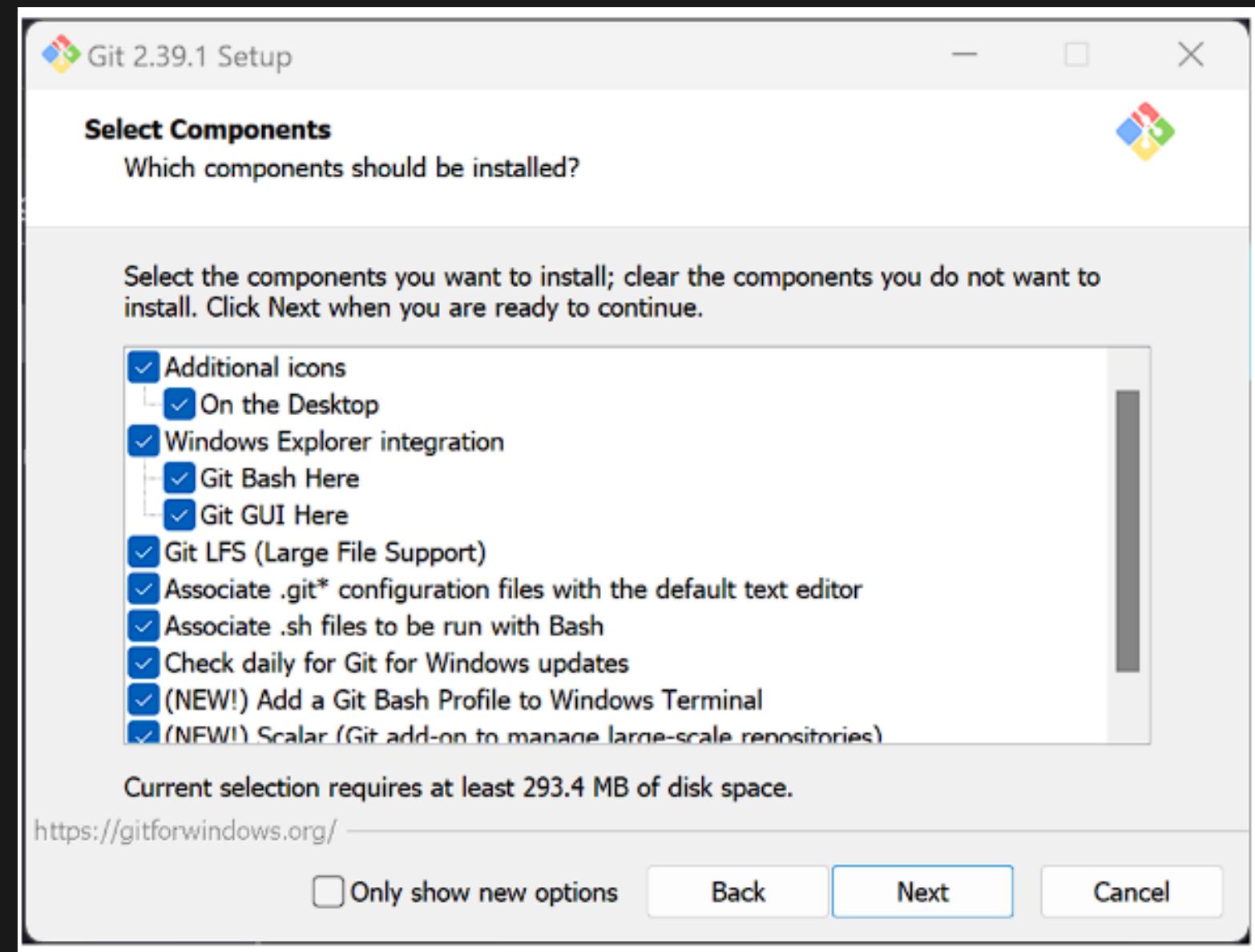
[Download for Windows](#)



Git Installation

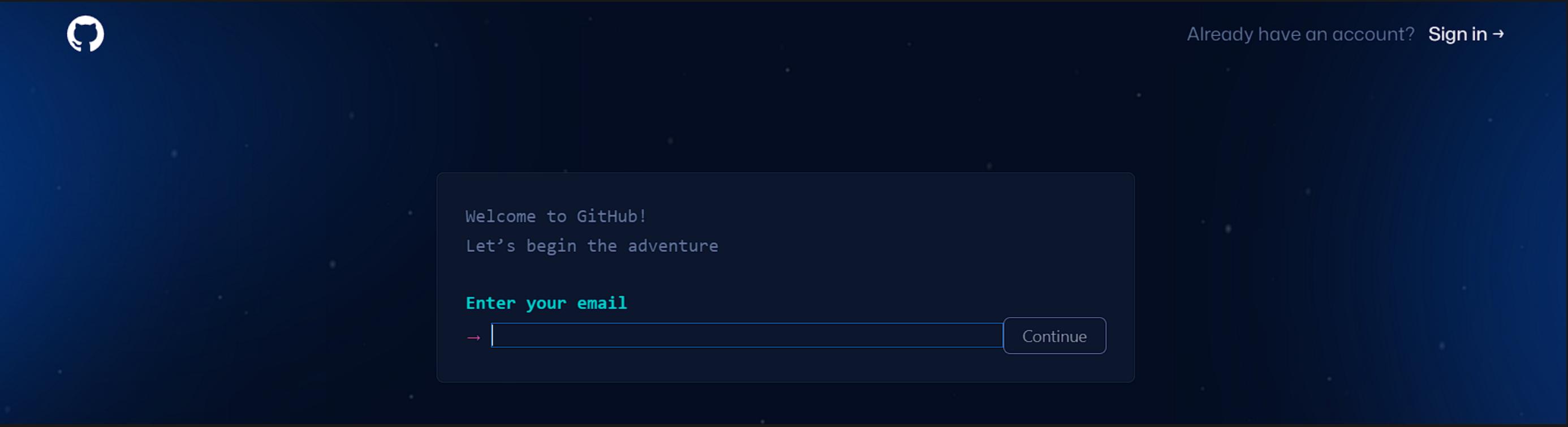
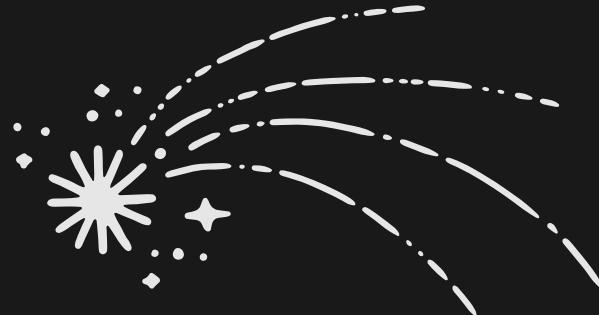


ติดตั้งตามขั้นตอน โดยใช้ default option





Git Sign Up



A screenshot of the GitHub sign-up page. The background is dark blue. At the top left is the GitHub logo (a octocat icon). At the top right is the text "Already have an account? [Sign in →](#)". In the center is a light blue rectangular form. It contains the text "Welcome to GitHub!" and "Let's begin the adventure". Below that is the instruction "Enter your email" in green, followed by an input field with a pink arrow icon and a "Continue" button.



Basic git operation

- Git เป็นโปรแกรม version control ที่นิยมใช้กันมาก หน้าที่ของ Git คือ เก็บ snapshot ของไฟล์ใน working folder ไปไว้ที่ repository
- ไปที่ desktop สร้าง folder ชื่อ my-project
- คลิกขวาที่ folder และเลือก Git Bash Here จะเปิดหน้าต่าง Git Bash
- ใช้คำสั่ง git init เพื่อเริ่มใช้ git ใน folder
- ให้ตั้ง config ของ folder ให้สามารถมองเห็น hidden
- จะพบว่ามีการสร้าง folder .git ขึ้นมา



Basic git operation

```
MewPk@DESKTOP-D5GFVOU MINGW64 ~/OneDrive/Desktop/my-project
$ git init
Initialized empty Git repository in C:/Users/MewPk/OneDrive/Desktop/my-project/.

MewPk@DESKTOP-D5GFVOU MINGW64 ~/OneDrive/Desktop/my-project (master)
$ ls

MewPk@DESKTOP-D5GFVOU MINGW64 ~/OneDrive/Desktop/my-project (master)
$ ls -al
total 12
drwxr-xr-x 1 MewPk 197121 0 Mar 30 17:00 .
drwxr-xr-x 1 MewPk 197121 0 Mar 30 17:00 ..
drwxr-xr-x 1 MewPk 197121 0 Mar 30 17:00 .git/

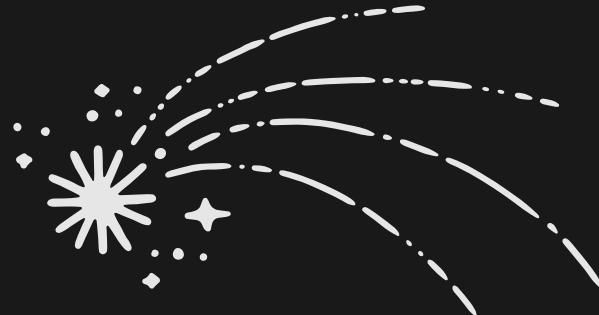
MewPk@DESKTOP-D5GFVOU MINGW64 ~/OneDrive/Desktop/my-project (master)
$
```

A screenshot of a Windows File Explorer window. The address bar shows the path: OneDrive - Personal > Desktop > my-project >. The main area displays a list of files and folders. There is a folder named ".git" which is highlighted with a yellow selection bar at the bottom. The columns are labeled Name, Date modified, Type, and Size. The size column is currently empty.

Name	Date modified	Type	Size
.git	3/30/2023 5:00 PM	File folder	



Basic git operation



- เริ่มด้วยการกำหนดผู้ใช้และ email โดยใช้คำสั่ง
 - `git config --global user.name <Name>`
 - `git config --global user.email <Email>`
- สามารถใช้คำสั่ง `git config -list` เพื่อตรวจสอบได้
(มีเครื่องหมาย – จำนวน 2 ตัว)



Basic git operation

- ให้สร้างไฟล์ขึ้นมา 2 ไฟล์ใน folder ชื่อ file1.txt กับ file2.txt
- คำสั่ง touch เป็นคำสั่งที่ใช้เปลี่ยน วัน-เวลา ของไฟล์ ถ้าใช้กับไฟล์ใหม่ จะเป็นการสร้างไฟล์ที่มีขนาด 0 ไบต์
- คำสั่ง echo เป็นค่าสั่งที่เอาข้อความที่อยู่ในเครื่องหมาย “ ” ไปสร้างไฟล์และใส่ในไฟล์ที่สร้างขึ้น



Basic git operation

```
MewPk@DESKTOP-D5GFVOU MINGW64 ~/OneDrive/Desktop/my-project (master)
$ touch file1.txt

MewPk@DESKTOP-D5GFVOU MINGW64 ~/OneDrive/Desktop/my-project (master)
$ echo "file2" > file2.txt

MewPk@DESKTOP-D5GFVOU MINGW64 ~/OneDrive/Desktop/my-project (master)
$ ls -al
total 13
drwxr-xr-x 1 MewPk 197121 0 Mar 30 17:04 .
drwxr-xr-x 1 MewPk 197121 0 Mar 30 17:00 ..
drwxr-xr-x 1 MewPk 197121 0 Mar 30 17:00 .git/
-rw-r--r-- 1 MewPk 197121 0 Mar 30 17:04 file1.txt
-rw-r--r-- 1 MewPk 197121 6 Mar 30 17:04 file2.txt
```



Basic git operation

- ใช้คำสั่ง `git status` ซึ่งจะแสดงสถานะของไฟล์ใน git
- จะแสดงให้เห็นว่าไฟล์ก็ 2 ไฟล์อยู่ในสถานะ untracked

```
MewPk@DESKTOP-D5GFVOU MINGW64 ~/OneDrive/Desktop/my-project (master)
$ git status
On branch master

No commits yet

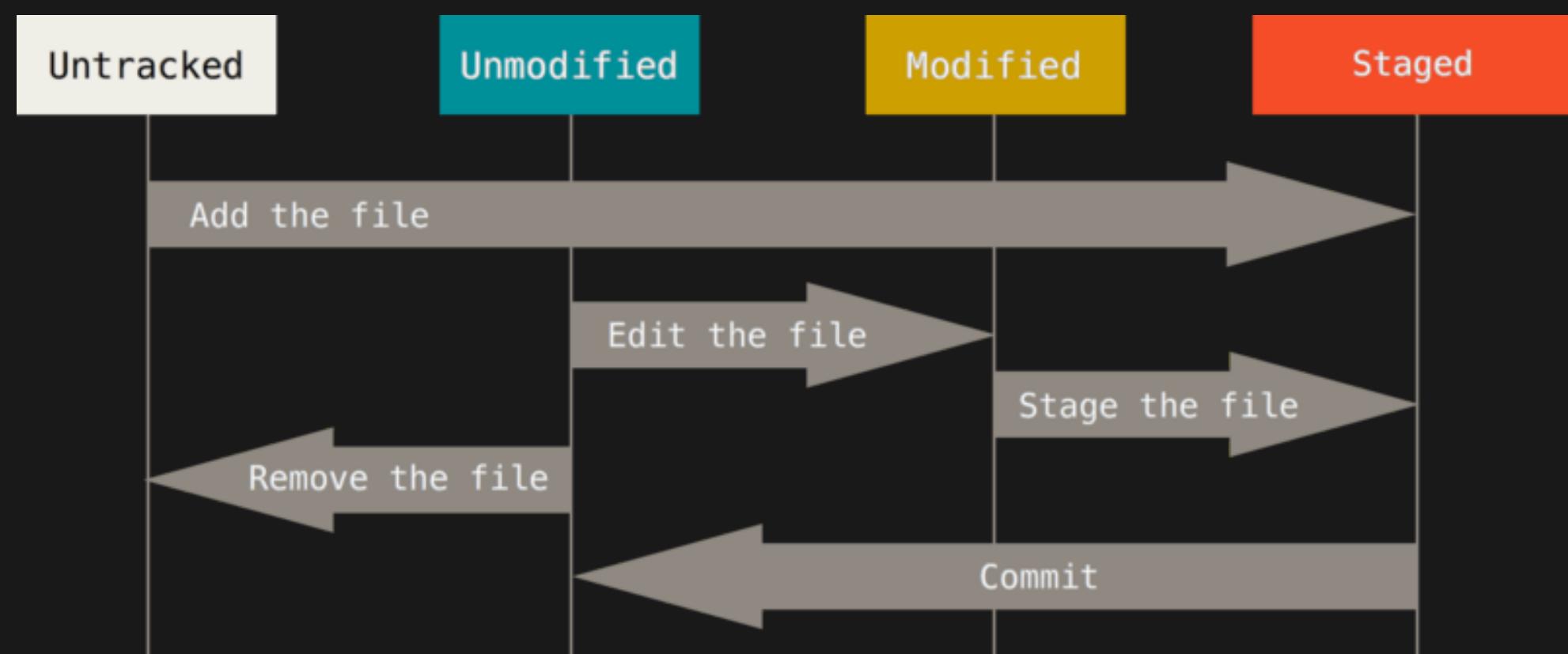
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    file1.txt
    file2.txt

nothing added to commit but untracked files present (use "git add" to track)
```



Basic git operation

- ไฟล์ใน git จะมี 4 สถานะ
 - untracked ยังไม่มีอยู่ใน “การติดตาม” ของ git
 - ถ้า add file เข้าสู่ git จะเข้าสู่สถานะ staged (“ติดตาม” แต่ยังไม่ทำ snapshot)
 - ถ้า commit จะทำ snapshot และเข้าสู่สถานะ unmodified
 - ถ้าไฟล์ที่ commit แล้ว มีการแก้ไขจะเข้าสู่สถานะ modified





Basic git operation

- ใช้คำสั่ง git add เพื่อนำไฟล์เข้าสู่สถานะ staged
- สัญลักษณ์ . มีความหมายว่าทุกไฟล์



Basic git operation

- ในขณะที่ไฟล์อยู่ในสถานะ staged หากมีการแก้ไข จะไปอยู่ในสถานะ modified
- ถ้าต้องการให้ไฟล์มาอยู่ในสถานะ staged ใหม่ ก็ให้ git add อีกครั้ง



Basic git operation

- จะใช้คำสั่ง git commit เพื่อนำไฟล์เข้าสู่ repository โดยคำสั่ง commit จะต้องใส่ข้อความกำกับด้วย เมื่อ commit แล้วจะเกิด snapshot ของไฟล์ที่ commit

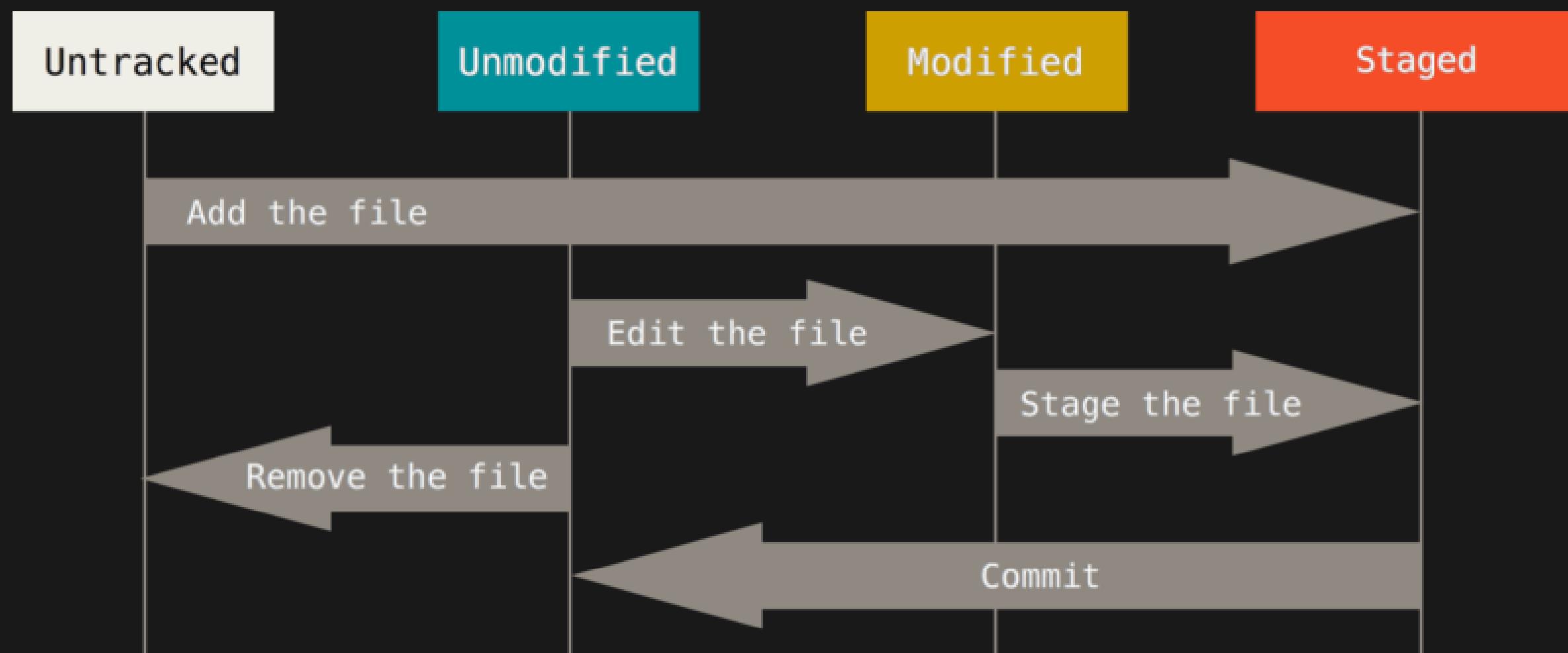
```
MewPk@DESKTOP-D5GFVOU MINGW64 ~/OneDrive/Desktop/my-project (master)
$ git add .
warning: in the working copy of 'file1.txt', LF will be replaced by CRLF the next time Git touches it

MewPk@DESKTOP-D5GFVOU MINGW64 ~/OneDrive/Desktop/my-project (master)
$ git commit -m "mewpk first commit!"
[master (root-commit) ab04872] mewpk first commit!
 2 files changed, 2 insertions(+)
  create mode 100644 file1.txt
  create mode 100644 file2.txt

MewPk@DESKTOP-D5GFVOU MINGW64 ~/OneDrive/Desktop/my-project (master)
$ git status
On branch master
nothing to commit, working tree clean
```



Basic git operation





Basic git operation

- เราสามารถใช้คำสั่ง git log ในการตรวจสอบ snapshot
- สมมติว่าเราสร้างไฟล์กี่ 3 ชื่อ file3.txt ไฟล์นี้จะอยู่ในสถานะ untracked

```
MewPk@DESKTOP-D5GFVOU MINGW64 ~/OneDrive/Desktop/my-project (master)
$ git log
commit ab0487236b0fd5cd4525a5c56ed305ee34b754de (HEAD -> master)
Author: MewPK <milzaza123@gmail.com>
Date:   Thu Mar 30 17:09:51 2023 +0700

    mewpk first commit!

MewPk@DESKTOP-D5GFVOU MINGW64 ~/OneDrive/Desktop/my-project (master)
$ touch file3.txt

MewPk@DESKTOP-D5GFVOU MINGW64 ~/OneDrive/Desktop/my-project (master)
$ git status
On branch master
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    file3.txt

nothing added to commit but untracked files present (use "git add" to track)
```



Basic git operation



สรุปคำสั่ง

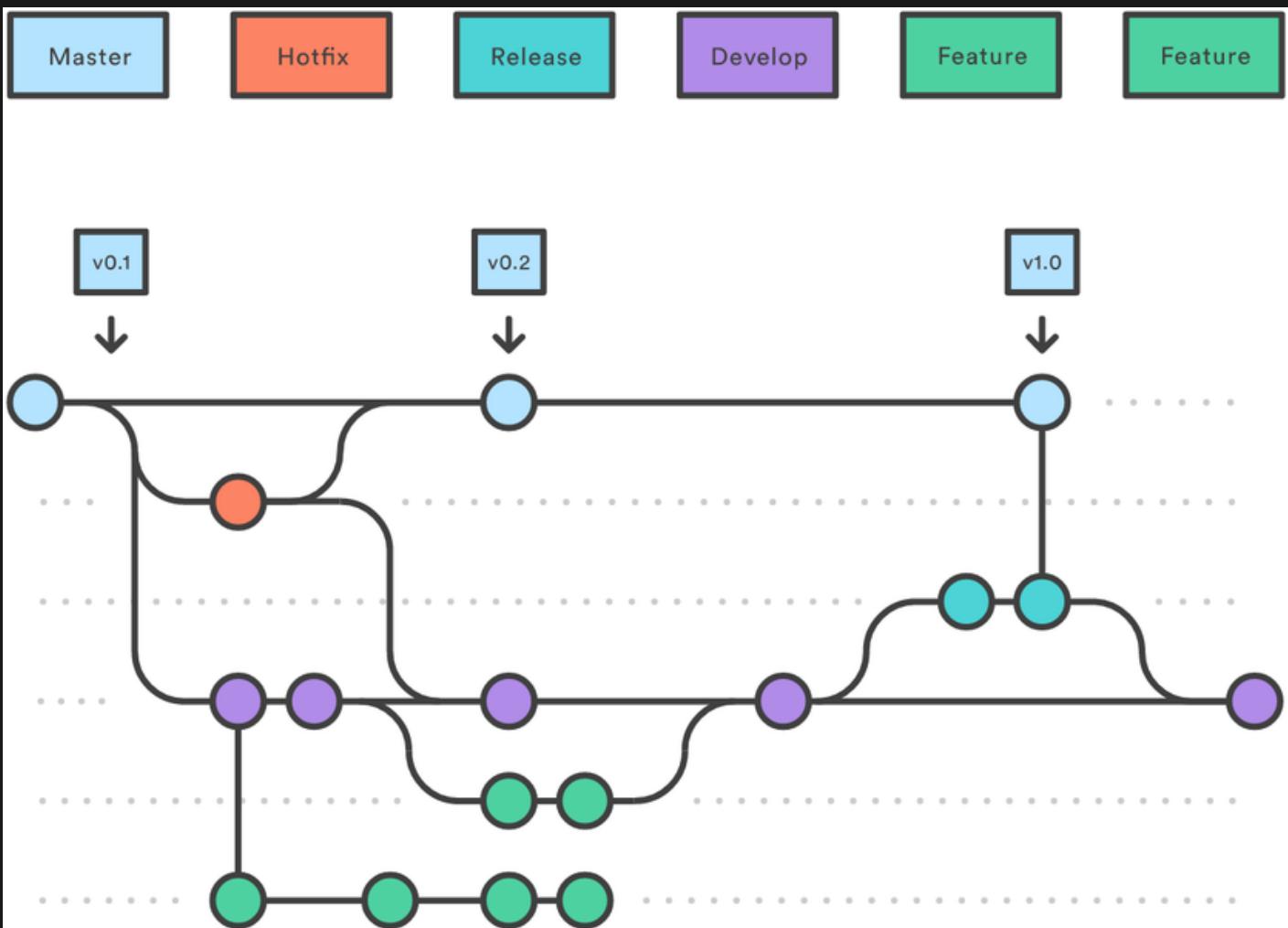
- `git config` กำหนดค่าต่างๆ เช่น ชื่อ, email
- `git add` เพิ่มไฟล์ลงใน Staging Area
- `git status` แสดงสถานะปัจจุบันของ Git repository
- `git commit` เจียนการเปลี่ยนแปลงลงใน Git repository
- `git log` ประวัติการเปลี่ยนแปลง
- `git checkout` เปลี่ยนไปยัง commit ต่างๆ
- `git reset` ย้อนกลับการ commit



Git Branch



- ในบางครั้งเราสามารถแยกเส้นทางการพัฒนาโปรแกรม เพื่อพัฒนาแต่ละ feature แยกกัน หรือร่วมงานกันหลายคน
- แต่ละเส้นทางจะเรียกว่า branch





Git Branch

- ในการสร้าง branch ให้มีจะมีทางเลือกดังนี้
 - คำสั่ง `git branch <name>` จะสร้าง branch ใหม่
 - คำสั่ง `git checkout -b <name>` จะสร้าง branch ใหม่ และเปลี่ยน default ไปยัง branch ที่สร้างใหม่

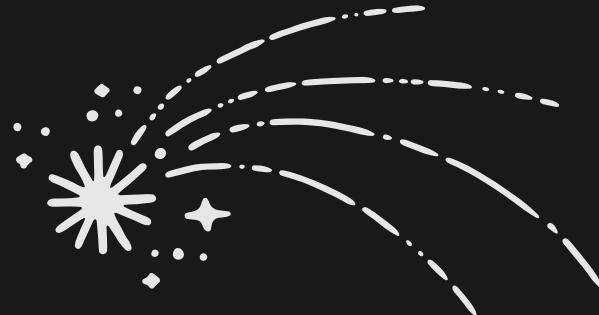


Git Branch





GitHub



- ให้เข้าไปที่ GitHub และสร้าง GitHub Account (หากยังไม่มี)
- วิธีที่ง่าย คือ สร้าง Repository บน GitHub จากนั้นค่อย Clone มาที่เครื่อง
- ให้กดเครื่องหมาย + กี่มุ่มบนขวา และ เลือก New Repository จากนั้นป้อนข้อมูลและกด Create repository



GitHub

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere?
[Import a repository.](#)

Owner *

 mewpk /

Repository name *

Great repository names are short and memorable. Need inspiration? How about [expert-guacamole](#)?

Description (optional)

 Public

Anyone on the internet can see this repository. You choose who can commit.

 Private

You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

Add a README file

This is where you can write a long description for your project. [Learn more](#).

Add .gitignore

Choose which files not to track from a list of templates. [Learn more](#).

.gitignore template:

Choose a license

A license tells others what they can and can't do with your code. [Learn more](#).

License:

 You are creating a public repository in your personal account.



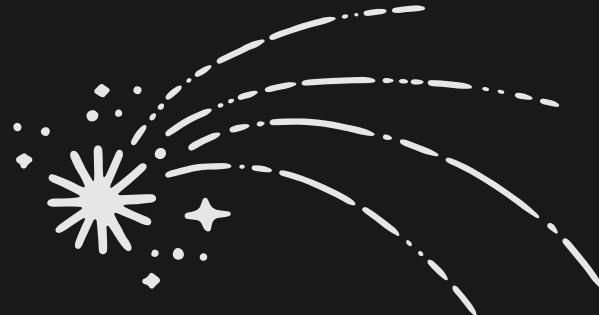
GitHub



- ให้คลิ๊ก link ในกรอบเอาไว้ จากนั้นไปที่ desktop หรือที่ต้องการ
จากนั้นเรียก command line
- แล้วใช้คำสั่ง `git clone <link>`



GitHub



Quick setup — if you've done this kind of thing before

Set up in Desktop

or

HTTPS

SSH

<https://github.com/mewpk/aaa.git>



Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).

...or create a new repository on the command line

```
echo "# aaa" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/mewpk/aaa.git
git push -u origin main
```



...or push an existing repository from the command line

```
git remote add origin https://github.com/mewpk/aaa.git
git branch -M main
git push -u origin main
```



...or import code from another repository

You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

[Import code](#)



GitHub



- `git push <remote> < branch >` เป็นการส่งไฟล์ที่ commit แล้วเข้าสู่ remote repository
- `git branch -a` บอกว่ามี branch ใดบ้าง รวมถึง remote ด้วย
- `git branch -vv` แสดง upstream branch และ commit ล่าสุด



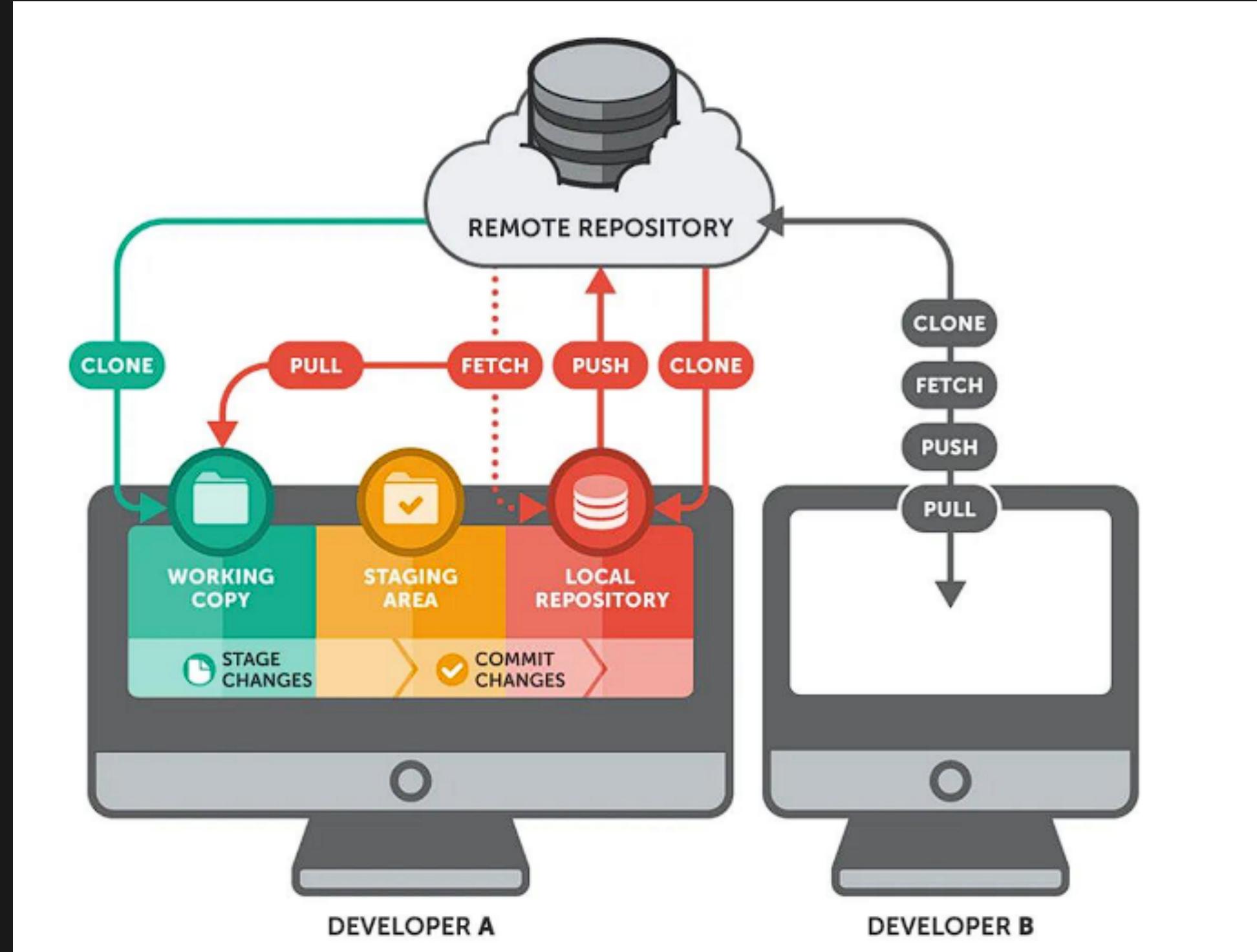
GitHub



- git fetch ใช้ตรวจสอบว่าไฟล์ใน local repository กับ remote repository มีความแตกต่างกันหรือไม่ ไฟล์ใหม่กว่า หรือ ล้าสมัยอยู่กี่ commit
- git pull เป็นการดึงไฟล์ที่มีการเปลี่ยนแปลงใน remote repository มาเปลี่ยนแปลงใน local work directory ซึ่งเทียบได้กับการใช้คำสั่ง git fetch ซึ่งเป็นการ update ในส่วน repository ของ git จากนั้นจึง ตามด้วยคำสั่ง git merge ซึ่งเป็นนำข้อมูลใน repository มา merge ลงใน local work directory ก็จะเห็นไฟล์ตรงกับใน remote repository



GitHub





GitHub



- Pull requests มีความหมายว่า GitHub มีการ update ใน branch อื่น ในการนี้ที่มีผู้พัฒนาหลายคนทำงานร่วมกันผ่าน GitHub และ ต้องการรวม feature ที่พัฒนาเข้ากับ main branch หรือ branch หลัก และ ต้องการ merge branch กลับมาที่ repository หลัก จึงขอความเห็น จากผู้พัฒนาอื่น ผ่าน Pull request และหากเห็นชอบจะได้ merge ต่อไป



GitHub



ให้จับกลุ่ม 2-3 คน

- ให้ตัวแทน 1 คน สร้าง repository และ เพิ่มผู้พัฒนาเข้าใน GitHub repository ให้เข้าไปที่ repository จากนั้นเลือก Setting จะเปิดหน้าใหม่ ให้เลือก Collaborators ให้เพิ่มผู้ใช้ในทีมเข้าไป จากนั้น GitHub จะส่ง invite mail ไป ให้ตอบรับ
- ให้คนในทีม สร้าง branch ของตัวเอง โดยเป็นชื่อของตนเอง
- โดยมีการเพิ่ม folder ของงานเก่าๆ ของเข้าไปใน branch
- หลังจากนั้นสร้าง Pull requests