



HTML & CSS

Beginner to Super Beginner

Author - Lwin Moe Paing

မိတ်ဆက်

ဒီစာအပ်က Web Development ကိုအခုံမှစလေ့လာမယ့်သူတွေအတွက် ရေးသားဖြစ်ခဲ့တာပါ။ ဒီစာအပ်ထဲမှာ ပါဝင်လာမယ့် အကြောင်းအရာတွေသည် Web Development ထဲမှာမှ မရှိမဖြစ်လိုအပ်တဲ့ HTML & CSS အကြောင်းအရာတွေပဲဖြစ်ပါတယ်။ သို့ပေါ်မယ့် တကာယ်လက်တွေ၊ Real World Project တွေမှာတော့ ဒီအကြောင်း အရာ (၂) ရတည်းနဲ့တော့ လုံလောက်မှာမဟုတ်ပါဘူး အခြားဆက်စပ်နည်းပညာတွေ အများကြီးလေ့လာစရာ ကျေနိုရိုနေပါသေးတယ်။ Web Foundation ကိုစလုပ်မယ့်သူတွေ နဲ့ အခုံမှစတင်လေ့လာမယ့်သူများက ဘယ်ကနေစရမလဲ မသိတဲ့အတွက် ဒီစာအပ်က များစွာ အထောက်အကျေပြွေစေလိမ့်မယ်လို့ လည်း ယုံကြည်ပါတယ်။ မဖြစ်မနေ ထပ်လေ့လာရမယ့် JavaScript ကိုတော့ ဒီစာအပ်ထဲမှာ ဖော်ပြသွားမှာမဟုတ်ပါဘူး။

စာရေးသူအကြောင်း

ကျွန်တော်ကိုတော့ ကိုလွှင်မိုးပိုင် လို့ခေါ်ပါတယ်။ ယခု စာရေးနေစဉ်ကာလမှာ ကျွန်တော်က Web Development ပတ်ဝန်းကျင်မှာ ကျင်လည်ခဲ့တာ (၅) နှစ်ကျော် ကာလသို့ရောက်ရှိခဲ့ပြီဖြစ်ပါတယ်။ ယခုလက်ရှိ Software House ကုမ္ပဏီတစ်ခုမှာ Senior Frontend Developer အနေနဲ့ လုပ်ကိုင်လျက်ရှိပြီး။ အပြင် Project ပေါင်းများစွာ ကိုလည်း လက်ခံရေးသားပေးလျက်ရှိပါတယ်။ ယခင်က ကျွန်တော် uab bank (2022) နဲ့ MoMoney (powered by ShweBank) တို့မှာ React Native Mobile Developer အဖြစ်လည်း တာဝန်ယူလုပ်ဆောင်ခဲ့ဖူးပါတယ်။ Marathon Myanmar Devliery Co, မှာလည်း Senior Vue Developer အနေနဲ့ 2020 တိုန်းက လုပ်ကိုင်ဆောင်ရွက်ခဲ့ပါတယ်။ UCSY ကို 2013 မှာ တက်ခဲ့ပြီး တဗ္ဗာသို့လ် ကျောင်းသားဘဝတိုန်းက ရရှိခဲ့တဲ့ Achievements တွေကတော့ Unihack Hackathon (2018) မှာ Winner နဲ့ Japan-Myanmar ပူးပေါင်းစီစဉ်သော Wit Design Award (2018) ကိုရရှိခဲ့တာဖြစ်ပြီး၊ နောက်ထပ် Wit Champion Award ကို (2020) မှာ ရရှိခဲ့ပါတယ်။ အခု စာအုပ်ကတော့ ကျွန်တော့ရဲ့ ပထမဆုံးရေးသားတဲ့ Ebook ပဲဖြစ်ပါတယ်။ ဒါကြောင့် ကိုယ့်ပါတ်ဝန်းကျင် Community မှာ အထောက်အကျေပြနိုင်မယ့် နောက်ထပ်စာအုပ်ပေါင်းများစွာကိုလည်း ရေးသားသွားချင်ပါသေးတယ်။

စာရေးသူအား ဆက်သွယ်လိုပါက - lwinmoepaing.dev@gmail.com (သို့)
ဖုန်းနံပါတ် - 09420059241 ကို ဆက်သွယ်နိုင်ပါတယ်။

လိုင်စင်

ဒီစာအုပ်ကို အခကြေးငွေဖြင့် ပြန်လည် ရောင်းချုပ်နှင့်များ ပြုလုပ်လို့မရပါ။

Credit ပေးပြီးတော့ ပြန်လည် Sharing လုပ်ခြင်း ပြုလုပ်နိုင်ပါသည်။

Non Commercial အဖျိုးအစားဖြစ်ပါသည်။ Orginal Same License အောက်မှာတော့

Contributions များလုပ်ဆောင်နိုင်ပါသည်။

အောက်ကလင့်မှာ Permission တွေအတွက် သွားရောက် ဖတ်လို့ရပါတယ်။

<https://creativecommons.org/licenses/by-nc-sa/4.0>

HTML & CSS: Beginner to Super Beginner © 2023
by Lwin Moe Paing is licensed under CC BY-NC-SA 4.0



Table of Contents

| | |
|---|----|
| Web တွေ ဘယ်လိုအလုပ်လုပ်သလဲ ? | 1 |
| နောက်ကျယ်က အကြောင်းအရာများ | 2 |
| Preparing for Coding | 5 |
| HTML for Dummies | 9 |
| HTML Elements | 17 |
| Heading Element များ <h1> to <h6> | 17 |
| Paragraph စာပိုဒ်များအတွက် <p> | 19 |
| Image များ ပုံများအတွက် | 21 |
| List များဖော်ပြရာအတွက် | 24 |
| Anchor Link လင့်များအတွက် <a> | 28 |
| From Theory to Practice: Your First HTML | 32 |
| Table အယားများတည်ဆောက်ခြင်း | 34 |
| Form Elements များ | 40 |
| Form Control များ | 41 |
| Form တွေ ဘယ်လိုအလုပ်လုပ်လဲ ? | 42 |
| Text Input တွေကိုလေ့လာကြည့်မယ် | 43 |
| Selector ခြေုံယူရာတွင်သုံးမယ့် Form Controls များ | 45 |
| Form Submit Element | 48 |
| Form Upload Element | 48 |
| From Theory to Practice - 2 : New HTML Practice | 49 |
| Formatting Elements များလေ့လာခြင်း | 51 |
| HTML Symbols များလေ့လာခြင်း | 52 |
| Generic Elements များဖြစ်တဲ့ <div> | 54 |
| Comment in HTML <!-- --> | 56 |
| HTML5 Layout Elements များ | 57 |
| CSS for Dummies | 61 |
| CSS ရေးသားဖွဲ့စည်းပုံကို လေ့လာခြင်း | 62 |

| | |
|--|------------|
| CSS အကြောင်းအရာများ | 65 |
| CSS ကို Inline Style အသုံးပြုခြင်း | 65 |
| CSS ကို Internal Style အသုံးပြုခြင်း | 67 |
| CSS ကို External Style အသုံးပြုခြင်း | 69 |
| CSS Selector ဆိတာ | 71 |
| CSS Universal Selector | 74 |
| CSS Type Selector | 76 |
| CSS ID Selector | 78 |
| CSS Class Selector | 80 |
| CSS Child Selector | 84 |
| CSS Descendant Selector | 87 |
| CSS Adjacent Sibling Selector | 89 |
| CSS General Sibling Selector | 91 |
| From Theory to Practice: CSS Selector | 94 |
| CSS နှင့် Colors | 97 |
| Colors တန်ဖိုးများ | 97 |
| RGB Color တန်ဖိုးအသုံးပြုခြင်း | 98 |
| Hexadecimal Color တန်ဖိုးအသုံးပြုခြင်း | 99 |
| Color Name တန်ဖိုးအသုံးပြုခြင်း | 101 |
| HSL Color တန်ဖိုးအသုံးပြုခြင်း | 101 |
| CSS Opacity ကိုလေ့လာခြင်း | 103 |
| CSS နှင့် Fonts များအကြောင်း | 109 |
| Font Family များသိခရီးထွက်ခြင်း | 109 |
| CSS နှင့် Typeface များအကြောင်း | 110 |
| CSS နှင့် Fonts များကို အသုံးပြုခြင်း | 112 |
| CDN fonts အသုံးပြုခြင်း | 114 |
| Custom fonts များအသုံးပြုခြင်း | 117 |
| Text Style များကို လေ့လာခြင်း | 121 |
| - Font Weight အသုံးပြုခြင်း | 121 |
| - Font Style အသုံးပြုခြင်း | 122 |

| | |
|---|------------|
| - Text တွေအတွက် Units များ | 122 |
| - Text Transform စကားလုံးတွေကိုပြောင်းလဲမယ် | 123 |
| - Text Decoration စကားလုံးများ | 124 |
| - Text Line-Height ကိုလေ့လာခြင်း | 126 |
| - Text Letter Spacing နဲ့ Word Spacing | 127 |
| - Text တွေကို Alignment ချိန်ခြင်း | 127 |
| - Text များအတွက် အရှင် | 129 |
| Attribute Selector များ နှင့် Pseudo Class | 133 |
| Pseudo-Classes များအကြောင်း | 133 |
| - Hyperlink <a> အတွက် Pseudo-Class များ | 133 |
| - Action အကြောင်းအနေကိုဖမ်းသော Pseudo-Class များ | 135 |
| Attribute Selector များအကြောင်း | 139 |
| CSS နှင့် Boxes များ | 143 |
| CSS Width များအကြောင်း | 147 |
| CSS Height များအကြောင်း | 154 |
| CSS Overflowing Contents များအကြောင်း | 159 |
| CSS Borders များအကြောင်း | 164 |
| CSS Border Radius အကြောင်း | 168 |
| CSS Margin အကြောင်း | 172 |
| CSS Padding အကြောင်း | 176 |
| CSS Box Shadow များအကြောင်း | 179 |
| From Theory to Practice: CSS Boxes | 187 |
| CSS List, Table များအကြောင်း | 189 |
| CSS List များအကြောင်း | 189 |
| CSS Table များအကြောင်း | 192 |
| CSS Position, Transition, Background Image | 195 |
| CSS Positions များအကြောင်း | 195 |
| CSS Transition များ | 202 |
| CSS Background Image အကြောင်း | 204 |

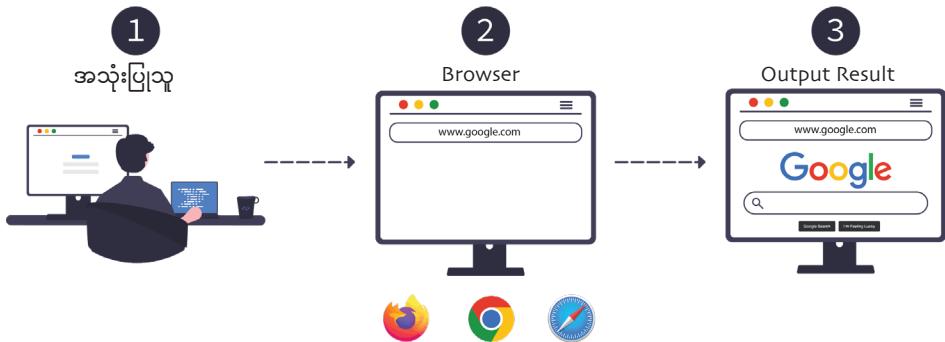
| | |
|---|-----|
| CSS ဖြင့် Layout များကို လေ့လာခြင်း | 207 |
| CSS Float အကြောင်း | 207 |
| CSS Grids အကြောင်း | 209 |
| - Grid Column ကိုစလေ့လာခြင်း | 210 |
| - Grid Rows ကို လေ့လာခြင်း | 213 |
| - Grid Gaps ကို လေ့လာခြင်း | 217 |
| - Grid Column Position နေရပါခြင်း | 219 |
| - Grid Alignment Items ချိန်ညွှန်ခြင်း | 221 |
| Grid System ကိုအသုံးပြုခြင်း | 227 |
| Responsive CSS ကိုလေ့လာခြင်း | 238 |
| Responsive ဆိတာဘာကြီးလဲ ဘာကြောင့်လေ့လာသင့်? | 238 |
| Responsive CSS ရေးထုံး ရေးသားနည်း | 239 |
| Responsive CSS Minimun Width ရေးသားနည်း | 241 |
| Responsive CSS Maximun Width ရေးသားနည်း | 245 |
| Responsive CSS Between ထဲရေးသားနည်း | 246 |
| Responsive Columns ရေးသားနည်း | 247 |
| Responsive LwinCafe | 249 |
| Flex Layout ကိုလေ့လာခြင်း | 249 |
| Flex ဘာလဲ ဘာလို့သုံးလဲ ? | 249 |
| Flex မှာဘာတွေပါလဲ လေ့လာခြင်း | 250 |
| Flex Container အကြောင်း | 252 |
| Flex Directions အကြောင်း | 252 |
| Flex Wrap အကြောင်း | 254 |
| Flex Flow အကြောင်း | 255 |
| Justify Content အကြောင်းအရာ | 255 |
| Align Items များအကြောင်းအရာ | 257 |
| Flex Gap များအကြောင်းအရာ | 258 |
| Flex Grow များအကြောင်းအရာ | 259 |
| Flex Align Self အကြောင်းအရာကိုလေ့လာခြင်း | 261 |
| နိဂုံးချုပ် | 262 |



ကျွန်ုတ်တို့ ၏ HTML&CSS - New Journey
ကို စတင်ပါတော့မယ် ...

Web တွေ ဘယ်လိုအလုပ်လုပ်သလဲ ?

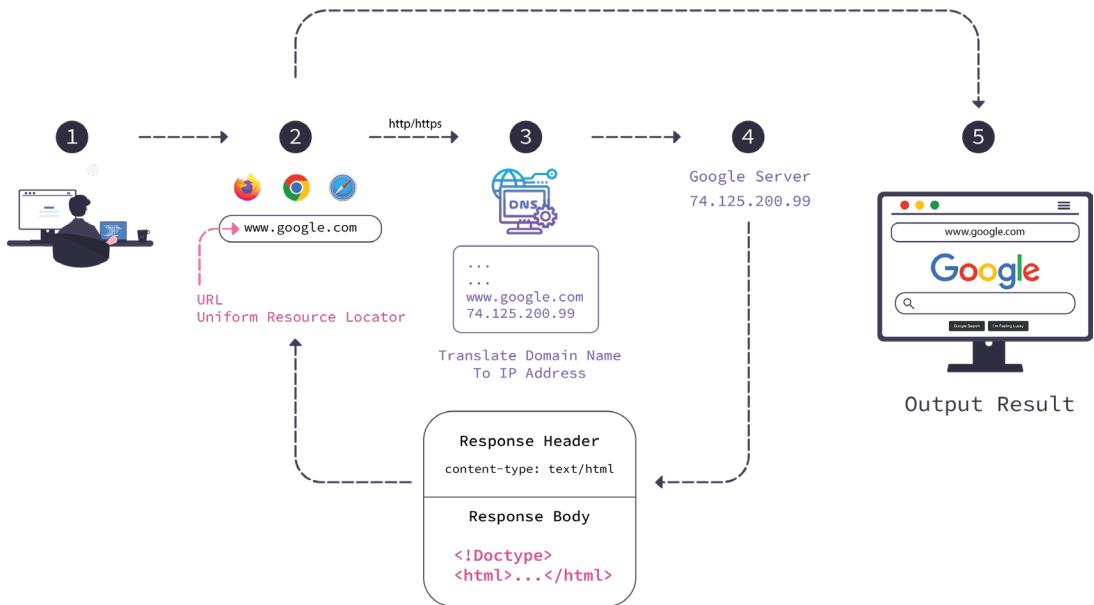
ကျွန်တော်တို့ အခါအသုံးပြုနေတဲ့ Google, Facebook, Youtube ကိုတို့ Browser ကနေ ပေါင်အသုံးပြုကြပါတယ်။ ဒီနေရာမှာ Browser ဆိုတာက Chrome, FireFox, Safari တို့ကိုပြောတာပါ။ အောက်က ပုံလေးကိုတစ်ချက်လေ့လာ ကြည့်ရအောင်



ကျွန်တော်တို့က နံပါတ် (1) ပုံက User တွေအသုံးပြုသူတွေပါ။ နံပါတ် (2) ပုံမှာရှိတဲ့ Browser တစ်ခုကိုသွားပြီး "www.google.com" ကို Input အဖြစ် Browser Bar မှာ ရှိက်ထည့်လိုက်တဲ့အခါ၊ နံပါတ် (3) ပုံလေးက အတိုင်း Google Search Engine Website ကို ကျွန်တော်တို့ အသုံးပြုနိုင်သွားမှာဖြစ်ပါတယ်။ ဒါပေါ်မယ် တကယ်နောက်ကွယ်က လုပ်ဆောင်ချက်တွေကတော့ အဲလောက်ရှိရှင်းနေတာမဟုတ်ပါဘူး။

အဲတော့ Big Picture မြင်ရဖို့အတွက် နောက်တစ်မျက်နှာကိုသွားလေ့လာရအောင်...

နောက်ကွယ်က အကြောင်းအရာများ



1 အရင်ဆုံး URL Bar မှာ “`www.google.com`” ဆိုပြီး ကျွန်ုတ်တို့က ထည့်လိုက်ပါတယ်။

မှတ်ချက်။ ။ ရိုက်လိုက်သော “`google.com`” ကို Domain Name လို့လည်းခေါ်ပါတယ်။ `google.com/maps` (or) `google.com/doodles` မှာ “/” နောက်က “`maps`” ရယ် “`doodles`” တို့ကို Path လမ်းကြောင်း လို့ခေါ်ပါတယ် Domain Name + Path ကောအပါအဝင်ကိုတော့ URL (Uniform Resource Locator) လို့ခေါ်ပါတယ်။

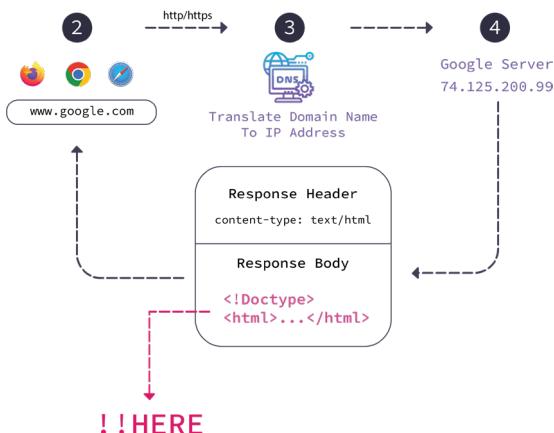
2 Browser ကနေတစ်ဆင့် Google Server ကိုခေါ်ပေးရမှာပါ သို့ပေါ်မယ် တကယ့် အပြင် လောကမှာ Server ကိုတိုက်ရှိက်မသိနိုင်ပါဘူး IP Address (Internet Protocol) အနေနဲ့ ပဲ သိနိုင်ပါတယ်။ အဲတော့ Browser ကနေ အရင်ဆုံး DNS Server ကိုသွားပါတယ်။

3 DNS (Domain Name System) ကိုဖြောတာပါ။ သူက Phone Numbers တွေမှတ်ထားတဲ့ စာအပ်နဲ့ဆင်တူပါတယ် ဥပမာ ကိုလွင်-09420059241 ပုံစံပေါ့ ကိုလွင် ဆိုတဲ့ နာမည် နှော့လိုက်ရင် ကျွန်ုတ်တော့ ဖုန်းနံပါတ်ကိုရမှာပါ။ အလားတူ DNS Server က Domain-Name တွေကို IP တွေနဲ့အတူတွဲမှတ်ထားပါတယ် ဘယ်လို့ Translate လဲဆိုရင် ခုကာပေးပို့လာတဲ့ `www.google.com` ရဲ့ တကယ့် IP Address အမှန်ကိုရှာပြီး Browser ကိုပို့လိုက်ပါတယ်။

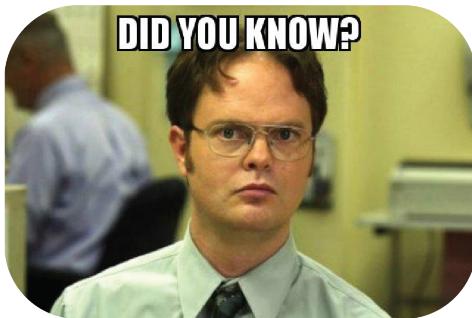
4 အဲဒီအခါ Browser က Google Server ဖြစ်တဲ့ IP 74.125.200.99 ကိုလုပ်း၏ပါတယ်။ ဒီနေရာမှာ ဘာနဲ့၏လဲဆိုရင် HTTP/HTTPS (HyperText Transfer Protocol) နဲ့၏ပါတယ်။ အထဲမှာ Meta Data တွေရယ် အခြား Data တွေလည်းအများကြီးပါပါသေးတယ်။ အခြေခံအားဖြင့် Request လုပ်တယ်လို့ မှတ်ယူထားပေးပါ။

5 Google Server က Request လုပ်လိုက်တာကို သိသွားတဲ့အတွက်ကြောင့် ကျွန်ုတ်တို့ Browser ကို Response အနေနဲ့ Data တွေပြန်လိုက်ပါတယ်။ အဲဒီအခြေအနေကို ပုံအောက်ဘက်က Box လေးကို တစ်ချက်ကြည့်ပေးပါ။ Response Header ထဲမှာ Content-Type အနေနဲ့ text/html ဆိုတာလေးကိုမြင်ရမှာပါ။ ဆိုလိုချင်တာက ပြန်လာတဲ့ Content က HTML ကြီးပါလို့ Browser ကိုအသိပေးလိုက်တာပါ။ ပြီးတော့ Response Body ထဲမှာတော့ ကျွန်ုတ်တို့ Code တွေကိုမြင်ရမှာပါ။

Browser က Response Body ထဲက HTML code တွေကို ဘာသာပြန်ပေးလိုက်တဲ့ အခါမှာ။ နံပါတ်(5) ပုံထဲကလို ကျွန်ုတ်တို့ Browser ထဲမှာ Output Result က မြင်သွားကြပါတယ်။



ဒီစာအုပ်လေးထဲမှာ လေ့လာသွားမယ့်အပိုင်းလေးက “!!HERE” ဆိုတဲ့ နေရာလေးတစ်ပိုင်းပါဖြစ်ပါတယ် ကျွန်ုတ်တို့က Browser က နားလည်တဲ့ HTML Code တွေကို ရေးသားလေ့ကြုင့်သွားကြမှာပါ။ ခြုံငံကြည့်ရမယ်ဆိုရင် Traditional Website တွေက အခြေခံအားဖြင့် Request , Response Method နဲ့ပါလုပ်လုပ်ကြလေ့ရှိတာကို မှတ်သားပေးထားပါ ခင်ဗျာ။



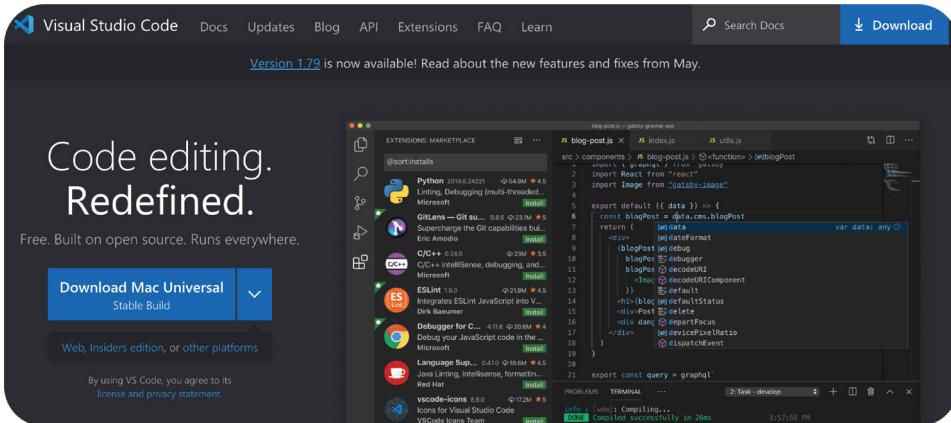
သင်သိပါသလား တော်တဲ့ Developer အများစုက
Single တွေဆိုတာ ??

Preparing for Coding

ကျွန်တော်တို့ စတင်ပြီး Code တွေမရေးခင် ဘာ Tool တွေသုံးသွားမလဲဆိတ္တာပြင်ဆင်ရပါတယ်။ ကျွန်တော်တို့ ရိုးရိုးတန်းတန်း Windows မှာဆို Notepad ရှိပြီး Mac မှာဆိုTextEdit တို့နဲည်းအသုံးပြုလုပ်ရပေါ်မယ် အား ဒီစာအုပ်ထဲမှာတော့ ကျွန်တော်က Code Editor အဖြစ် VS Code ကိုအသုံးပြုသွားမှာပါ။

- <https://code.visualstudio.com>

မှာ Download ဆွဲကာ Installation လုပ်ပြီးအသုံးပြနိုင်ပါတယ်။

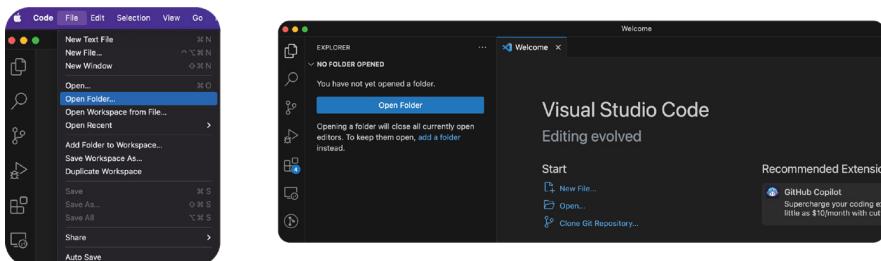


VS Code ကိုဖွင့်ပါ ပြီးနောက် လိုအပ်သော Extensions များကို အသုံးပြုရန် အောက်ပါအဆင့် 4 ခုအတိုင်း Download ပြုလုပ်ပါမည်။

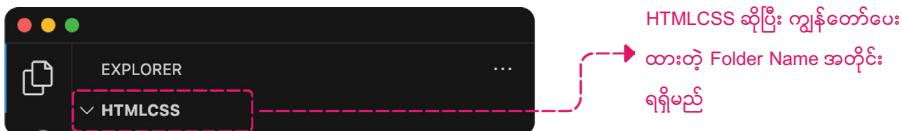


လိုအပ်တဲ့ Extensions တွေထည့်သွင်းပြီးတဲ့အခါ ပထမဆုံး ကိုယ်ပိုင် Folder လေးတစ်ခု တည်ဆောက်လိုက်ပါမယ်။ ကျွန်တော်ကတော့ Folder နာမည်ကို "HTMLCSS" လို့ ပေးလိုက်ပြီးတည်ဆောက်လိုက်ပါမယ်။ စာဖတ်သူတို့က ကြိုက်တဲ့ နာမည်ပေးပြီး ကိုယ်အဆင်ပြေရာ နေရာ(Directory) မှာတည်ဆောက်နိုင်ပါတယ်။

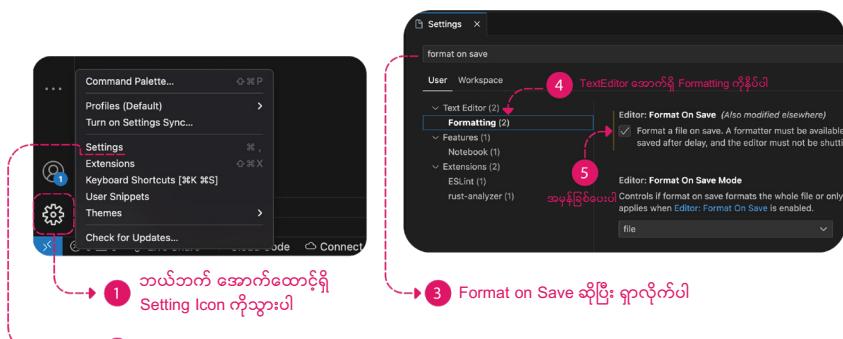
ပြီးနောက် အောက်ကပုံထဲကအတိုင်း Open Folder ဆိုပြီး ကိုယ်တည်ဆောက်ခဲ့တဲ့ Folder ထဲကိုဝင်ကြည့်မယ်ခင်ဗျာ။ ပုံ (၁) အရ အပေါ်ဘားထဲမှာ File ဆိုတာလေးကိုမြင်ရမှာပါ နှင့်ပြီး Open Folder ထဲကသွားလို့ရပါတယ် (သို့) ပုံ (၂) ထဲကအတိုင်းလည်း စစ်ဖွင့်ခြင်းမှာ Open Folder ဆိုတဲ့ အပြောရောင် Button လေးကိုနှင့်ပြီးလည်း သွားနိုင်ပါတယ်။ နှင့်ပြီးသွားတဲ့ အခါမှာတော့ ကိုယ်တည်ဆောက်ခဲ့တဲ့ Folder ကိုရွေးချယ်ပေးပါ။



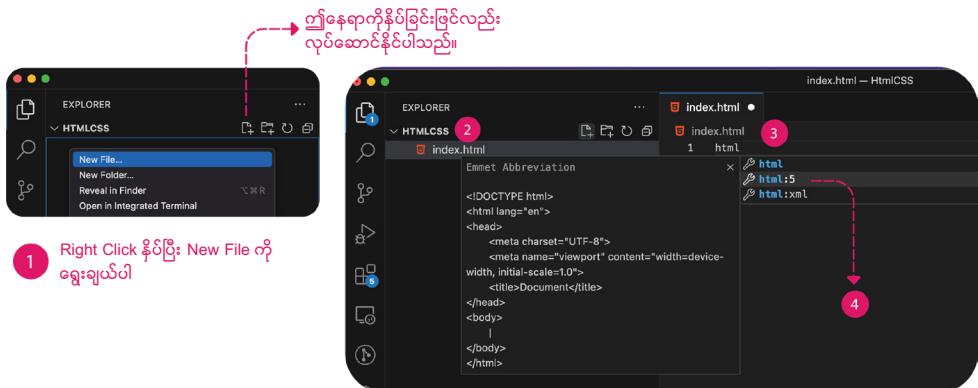
Folder ဖွင့်ဝင်သွားပြီးရင် အောက်ကပုံအတိုင်း ကိုယ်ပေးခဲ့တယ့် Folder နာမည်လေးက Explorer Bar မှာပေါ်နေမှာဖြစ်ပါတယ်



Setting ထဲမှာသွားပြီး Code တွေ Save တာနဲ့ AutoFormat ဖြစ်အောင်လုပ်ပါမယ်။ မသုံးလည်းရပါတယ် Beginner တွေအနေနဲ့ ပိုပြီး Prettier ဖြစ်အောင်လုပ်ချင်လိုပါ။



ကျွန်တော်တို့ ဘယ်ဘက် Explore ဘက်မှာ အောက်က ပုံအတိုင်း "index.html" ဆိုတဲ့ ဖို့လေးတစ်ခုတည်ဆောက်လိုက်ပါမယ်။



အဆင့် (၁) အနေနဲ့ File ကိုတည်ဆောက်ရန် Right Click နိပ်ပြီး New File လေးကို ရွေးချယ်လိုက်ပါ။ ပြီးနောက် အဆင့် (၂) "index.html" ဆိုပြီး ဖို့လေးတစ်ခုတည်ဆောက်ထဲမှတ်ပေးပါ။ အဆင့် (၃) ပုံလေးထဲကအတိုင်း ညာဘက်အခြမ်းတွင်ကျွန်တော်တို့ ပေးထားသော ဖိုင်(File) နာမည်နဲ့ Text Editor လေးပေါ်နေပါမည် အဲအတွင်းတွင် html ဆိုပြီး စာသားလေးရေးလိုက်ပါက

- html, - html:5, - html:xml

ဆိုပြီး Auto Suggest လုပ်ပေးထားသော Emmet Abbreviation လေးတွေပေါ်လာပါ မည် html:5 ကိုရွေးချယ်ပေးပါခင်ပျော်။ အောက်က ပုံအတိုင်း ပထမဆုံး ကျွန်တော်တို့၏ HTML Code လေးတို့ရရှိပါပြီ။ အခုနှစ် ကျွန်တော်တို့၏ Code Editor ပြင်ဆင်မှုပြီး ပြီးဆုံးသွားပါပြီ Congratulation ပါယွှေ့။ ဒါဆို နောက်ထက်ကျွန်တော်တို့ အောက်တွင်မှတ်ဆောင်ရွက်ပါ။

```
index.html

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Document</title>
  </head>
  <body></body>
</html>
```



HTML



HTML + CSS

HTML ဆိတာ Website ရဲ့ Foundation ပါပဲ
CSS လေးသာပါလိုက်ရင်တော့ ပိုပြီးလှပသွားမှာပါ။

HTML for Dummies



Tim Berners-Lee

HTML ဟာ Markup Language အမျိုးအစားဖြစ်ပြီး HTML ရဲ့ အရှည်ကတော့ HyperText Markup Language ဖြစ်ပါတယ်။ သူ့ကို Tim Berners-Lee ဆိုတဲ့ Computer Scientist က ပထမဆုံး ဖန်တီးခဲ့တာပါ။ ပြီးတော့ World First Browser ကိုလည်း ဖန်တီးခဲ့သူတစ်ယောက်ဖြစ်ပါတယ်။

အခြား Markup Language အများစုက ဘယ်လိုအလုပ်လုပ်သလဲဆိုရင် ကျွန်တော်တို့ ပေးလိုက်တဲ့ စာသား(Text) များကို Structure ကျစစ်ဖြင့်ရေးပြီး Computer ကို နားလည် အောင်ပြုလုပ်စေတာဖြစ်ပါတယ်။

Web Development မှာကျ HTML နဲ့ ဖော်ပြချင်တဲ့ စာသား(Text) တွေ (သို့) Content တွေကို Content Structure ပုံစံကို တည်ဆောက်စေပြီး Browser ကိုနားလည်အောင် ပြုလုပ်ရတာဖြစ်ပါတယ်။ အောက်က ပုံလေးကို လေ့လာကြည့်ရအောင်...



အဓိဋ္ဌာန်း: (၁) က ကျွန်တော်ရဲ့ Profile ကိုထည့်ထားတာမှန်းသိမှာပါ။

အဓိဋ္ဌာန်း: (၂) က နာမည်ပါ အဲဒါကို ခေါင်းစဉ်မှုလို့ စာသားအထူနဲ့ဖြင့်ရပါတယ်။

အဓိဋ္ဌာန်း: (၃) က Career ပါ သူက ခေါင်းစဉ်အသေးဖြစ်သွားပါပြီ။

အဓိဋ္ဌာန်း: (၄) က ကျွန်တော်အကြောင်းအကျဉ်းချုပ်ပါ အဲကောင်က စာကိုပုံစံပေါ့။

အဓိဋ္ဌာန်း: (၅) က တော့ TabBar မှာ နာမည်ရေးထားတာပါ။

ဒါတွေအကုန်လုံးကို ကျွန်တော်တို့က မျက်လုံးလေးနဲ့ကြည့်ရနဲ့ သိနိုင်ပါတယ် ဒါပေါ်မယ် Web Browser အနေနဲ့ဆို အဲလိုမသိနိုင်ပါဘူး Browser က HTML Code တွေကိုသာ သိမှာဖြစ်ပါတယ် အဲတော့ ကျွန်တော်တို့ ဘယ်လိုရေးသားထားလည်းဆိုတာ လေ့လာကြပါမယ်။

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<title>Lwin Moe Paing | Profile</title>
</head>
<body>


<h1>Lwin Moe Paing</h1>

<h3>React Native, Frontend Developer</h3>

<p>
  Hi, I'm Lwin Moe Paing. <br />
  ▶ I'm seeking assistance with the JavaScript language in order to develop
  numerous free software solutions to aid individuals who cannot afford to
  pay! <br />
  ◉ Feel free to ask me questions about React, React Native, and Node.js.
</p>
</body>
</html>

```

5

1

2

3

4

အခုပုံထဲတွင် HTML Element Tag များကို အပြောရောင်ဖြင့် ဖော်ပြထားသည့်ကို ပြင်ရပါမည်။

အညွှန်း (၁) က Profile ပုံလေးအတွက် HTML နဲ့ လေးရေးထားတယ်။

အညွှန်း (၂) က နာမည်ပါ အဲဒါကို ခေါင်းစဉ်မို့လို့ <h1> နဲ့ ရေးတယ်။

အညွှန်း (၃) က Career ပါ သူက ခေါင်းစဉ်အသေးဖြစ်လို့ <h3> နဲ့ရေးတယ်။

အညွှန်း (၄) က ကျွန်ုင်တော်အကြောင်း အကျဉ်းချုပ်ပါ <p> နဲ့ရေးပါတယ်။

အညွှန်း (၅) က <head> ထဲမှာရှိပြီး <title> ကိုအသုံးပြထားပါတယ် သူက TabBar ထဲ မှာသွားပေါ်မှာပါ။

Web Browser အနေနဲ့ဆို ဒါ HTML Code တွေကို သူက ဘာသာပြန်ပြီးခုကာက ကျွန်ုင်တော်ရဲ့၊ အကျဉ်းချုပ်ပြထားတဲ့ Web Page လေးကိုဖြင့်နိုင်အောင် ပြုလုပ် ပေးလိုက်တာဖြစ်ပါတယ်။ အခု အပေါ်က Code တွေကို သိစရာမလိုဘေးပါဘူး။ ကျွန်ုင်တော် တစ်ဆင့်ခြင်းရှင်းပြပေးသွားမှာပါ။

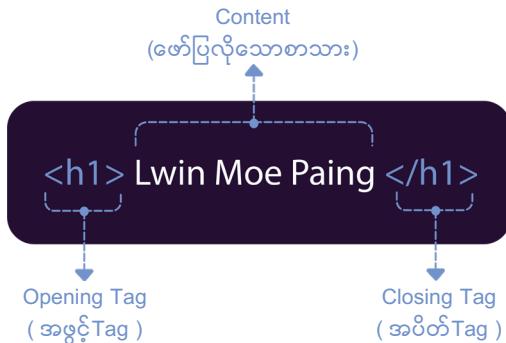
အခု Code တွေကို Run ချင်ရင်တော့ အောက်က Link ထဲသွားရောက် စမ်းနိုင်ပါတယ်။

<https://codepen.io/lwinmoeapaing/pen/YzRpjaV>

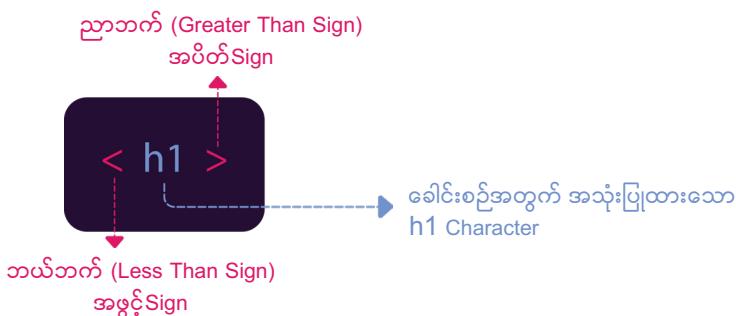
Code ကို ကြည့်ချင်ရင်တော့ အောက်က Link ကနေဝင်ကြည့်နိုင်ပါတယ်။

<https://gist.github.com/lwinmoeapaing/db12aabcf74c7e05a713a92ab78a0d65>

Web Page ရဲ့ Structure ကိုဖော်ပြန်အတွက် HTML က Element တွေကို အသုံးပြုကြပါတယ်။ Element အမျိုးအစားများစွာရှိပါတယ် ဥပမာ ခေါင်းစဉ်တွေ အတွက် `<h1>` ကိုအသုံးပြုသလို Paragraph စာပိုဒ်တွေအတွက် `<p>` ကိုအသုံးပြုပါတယ်။ မှတ်ရမှာက Element တော်တော်များများမှာ Opening Tag (အဖွင့်Tag) နဲ့ Closing Tag (အပိတ်Tag) ရှိကြပါတယ်။ အောက်ကပ်လေးကို လေ့လာကြည့်ရအောင် ...



ဒီထင်ပိုပြီး Opening Tag (အဖွင့်Tag) ကိုလေ့လာကြည့်မယ်ဆိုရင် ...



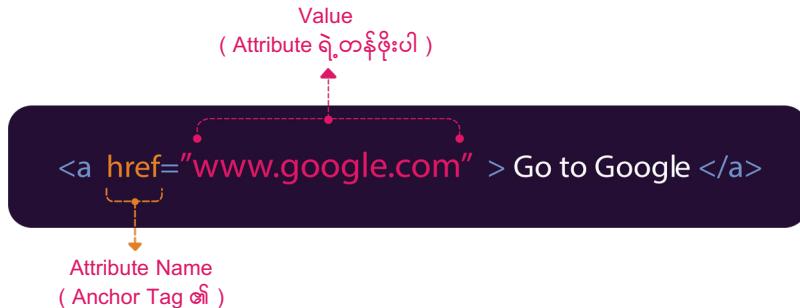
Closing Tag (အပိတ်Tag) ကို ကြည့်မယ်ဆိုရင်



ဖော်ပြပါ ဥပမာဏူးတွင် Opening Tag (အဖွင့်Tag) နှင့် Closing Tag (အပိတ်Tag)

ကြားထဲတွင် ရေးသားသောစာများကို Content ဟုခေါ်ပါတယ်။ သို့ပေါ်ယုံ

Opening Tag (အဖွင့်Tag) များထဲတွင် အခြားသော ဆိုင်ရာ Attributes များရှိနိုင်လေ သည်။ အဲဒါကို အောက်ပါပုံတွင် လေ့လာနိုင်သည်။



အခု ပေးထားတဲ့ Element မှာ Attribute Name (နာမည်) က href ဖြစ်ပြီးတော့

Attribute Value ကတော့ "www.google.com" ပါ။ အသုံးပြုထားတဲ့ `<a>` က Anchor ရဲ့ အတိုကောက်ကို ယူထားတာဖြစ်ပြီး သူကတော့ Link တွေချိတ်ဆက်ပေးတဲ့နေရာမှာ အသုံးပြုပါတယ်။ ဥပမာ အပေါ်က ရေးသားထားသောပုံစံသည် Go To Google ဆိုတဲ့ စာသားလေးကို နိပ်လိုက်ပါက `www.google.com` ကို လင့်ခဲ့ (link) လုပ်လိုက်ပြီး Google စာမျက်နှာကို သွားပါမယ်ဆိုတဲ့ အမိမိဖြစ်ပြီပါတယ်။ ကျွန်တော် Element တစ်ခုချင်းဆိုကို နောက်စာမျက်နှာများမှာ အသေးစိတ်ရှင်းလင်းသွားပါမယ် အခု ကတော့ Attribute တွေက Opening Tag ထဲမှာ ဒီလိုရေးသားသည်ကိုပဲမှတ်ထားပေးပါ။

ကျွန်တော်တို့ ဒီလိုပုံစံကို ပထမဆုံး Web Page ကိုအညွှန်းလေးတွေနဲ့ ပြတုန်းက Profile Picture ပုံလေးပေါ်အောင် အသုံးပြုခဲ့တဲ့ `` tag လေးကိုတစ်ခုကျက်သွားကြည့်နိုင်ပါတယ်။ သူ့မှာ src ဆိုတဲ့ Attribute Name နဲ့ ပုံရဲ့ တည်နေရာ url ကို Attribute Value အနေနဲ့ထည့်ပေးလိုက်ခြင်းဖြင့် Web Page မှာ ပုံလေးပေါ်လာခဲ့တာလည်း ဖြစ်ပါတယ်။

အခုဆုံး ကျွန်တော်တို့ အကြမ်းဖျဉ်းအားဖွင့် HTML element တွေရဲ့ ရေပုံရေးနည်းကို သိသွားကြပါပြီ။ အဲတော့ ကျွန်တော်တို့ စတင်ပြီး Element Tag တွေနဲ့ စာဖတ်သူတို့ရဲ့ စက်တွေမှာ လက်တွေ့ရေးသား လေ့ကျင့်ကြရအောင်...

Preparing for Coding အပိုင်းမှာ HTMLCSS ဆိုတဲ့ Folder နေရာတည်ဆောက်ပြီး ကျွန်တော်တို့ "index.html" ဆိုတဲ့ HTML file (စိုင်) လေးလုပ်ဖြစ်ခဲ့ပါတယ်။ အဲဒါမှာ html:5 ဆိုတဲ့ Auto Suggest လုပ်ပေးထားသော Emmet Abbreviation ကိုရွေးချယ်ခဲ့ဖြစ်ခဲ့တယ်။ အောက်ပါပုံလေးကို လေ့လာကြည့်ရအောင် ...



```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Document</title>
  </head>
  <body></body>
</html>
```

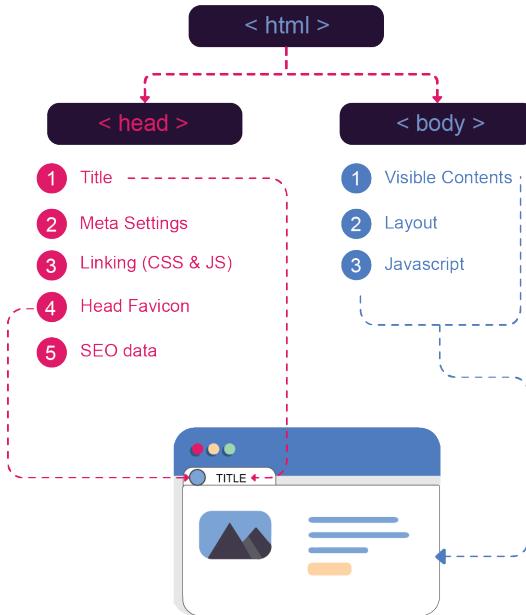
ပထမဆုံး စာကြောင်းမှာ <!DOCTYPE html> ဆိုတာ Document Type Declaration လုပ်တယ်လို့ခေါ်ပါတယ်။ Browser အနေနဲ့ အဲ Element ကိုမြင်တာနဲ့ HTML5 ရေးထုံး တွေနဲ့ ရေးသားထားမှန်း အသိပေးလိုက်တာပါ။

ဒုတိယအနေနဲ့ <html> tag ကိုမြင်ရမှာပါ ကျွန်တော်တို့ Attribute အကြောင်းကိုဖြောပြီးသားဖြစ်ပါတယ်။ ဒီနေရာမှာ "lang" က Attribute Name ဖြစ်ပြီးတော့ Attribute Value က တော့ "en" ဖြစ်ပါတယ်။ ဆိုလိုချင်တာက ဒီ HTML ရဲ့ content တွေရဲ့ အဓိက Language English ဖြစ်ပါတယ်လို့ Browser ကိုအသိပေးလိုက်တာပါ။

အခါက စပြီးတော့ ကျွန်တော်တို့က <html> tag အတွင်းထဲမှာ ၂ ပိုင်းရှိတာကို သတိပြုရပါယ်။ <head> နဲ့ <body> ပါ။ ၂ ပိုင်းလုံးကို <html> tag ရဲ့အတွင်းဘက်မှာရေးထားတာကို သတိပြုပေးပါ။

<head> အတွင်းပိုင်းမှာ ရေးသားတဲ့ Element အများစုက SEO (Search engine optimization) အတွက်ရေးကြတာဖြစ်ပြီး၊ ယခု HTML Document အတွက် Setting ထွေကို ကြိုသတ်မှတ်ထားတဲ့ ပုံစံမျိုးနဲ့ ရေးကြတာဖြစ်ပါတယ်။

<body> အတွင်းပိုင်းမှာ ရေးသားတဲ့ Element အများစုကတော့ Browser မှာ User ထွေမြင်ရတဲ့အပိုင်း(Visible Content)တွေကို ဖော်ပြန့်အတွက် ရေးကြပါတယ်။



<head> အတွင်းထဲမှာဆို အညွှန်း(၁) ရယ် အညွှန်း(၄) ရယ်အရ Browser ရဲ့ Tabbar မှာမြင်ရမယ့် ခေါင်းစဉ် (Title) ရယ် Logo Icon ကိုသတ်မှတ်ပေးလို့ရပါတယ်။ အဲ ၂ ခု ကပဲ Visible Contents ဖြစ်သွားတာပါ။

ကျွန်တဲ့ Element အများစုံက Meta Tag တွေပါပါမယ်။ သူက Browser ရဲ့ Setting ကို Configure ချေသလိုမျိုးပါ ဥပမာ <meta charset="UTF-8" /> ဆိုတဲ့ Element ကိုမြင်ခဲ့ရပြီးဖြစ်ပါတယ် သူရဲ့ အဓိပ္ပာယ်က ဒီ HTML Document အတွက် Character Encoding ကို Unicode UTF-8 Encoding System အဖြစ်သတ်မှတ်ပါတယ်ဆိုတဲ့ သဘောပါ။ အဲလိုမျိုး Meta tag တွေကို နောက်စာမျက်နှာတွေမှာ ကျွန်တော်ဖော်ပြုပေးသွားပါမယ်။

နောက် အညွှန်း(၃) အရ Linking (CSS & JS) ဆိုတာက ဒီ HTML အတွက် ပြင်ပ (external resources) သို့မဟုတ် အတွင်း (internal resources) မှာရှိတဲ့ CSS ဖိုင်တွေ Javascript ဖိုင်တွေရယ်ကို ချိတ်ဆက်ပေးတာမျိုးတွေကို <head> ထဲမှာရေးလေ့ရှိပါတယ်။ အညွှန်း(၅) အရကတော့ Google, Bing တို့လို့ ရှာဖွေရေး Website တွေက ကိုယ့်ရဲ့ Page ကိုလာရှာတဲ့အခါ သူတို့အတွက် နားလည်လွယ်အောင် Content တွေပြန်ပြရာမှာ သုံးတဲ့ Element အမျိုးအစားတွေကိုလည်း <head> ထဲမှာရေးပါတယ်။

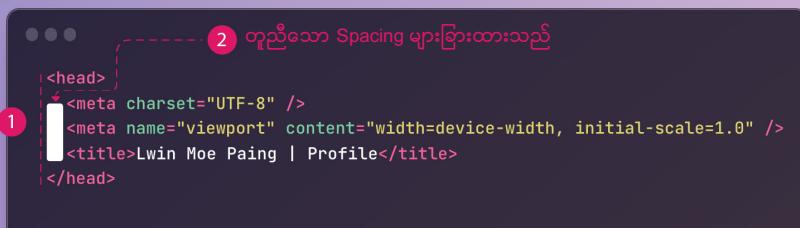
ဒါဆိုရင် <head> ထဲမှာရေးတဲ့ပုံစံတွေကို အကြမ်းဖျဉ်းနားလည်သွားပြီလို့ယူဆပါတယ် အခုအချိန်မှာ နားမလည်သေးလည်းကိစ္စမရှိပါဘူး စာဖတ်သူတို့ Code ရင်းနဲ့ သဘောတရားကို နားလည်လာပါလိမ့်မယ်။

<body> tag ထဲမှာတော့ ပုံလေးထဲကအတိုင်း Browser မှာ ပုံတွေ စာတွေ ခလုတ်လေး တွေကို ဖော်ပြထားတဲ့ Content တွေကို အစိကရေးသားရမှာဖြစ်ပါတယ်။ ကျွန်ုတ်တော် တို့ ဖော်ပြခဲ့ပြီးသော <h1>, <p>, , <a> များကို <body> tag အတွင်းပိုင်းမှာပဲ ရေးသားကြရမှာပါ။ Browser က အဲဒီ HTML Element တွေကို ဘာသာပြန်ပြီး User မြင်တွေရသောအပိုင်း (Visible Contents) အဖြစ်ဖော်ပြသွားမှာဖြစ်ပါတယ်။

မှတ်ချက်။ ။ ။ အချို့သော javascript , <script> tag တွေကိုလည်း <body> ထဲတွင် ထည့်သွင်းရေးသားလေ့ရှိပါသည်။

သတိပြုရမယ့်အချက်တစ်ချက်လည်းရှိပါသေးတယ် ကျွန်ုတ်တို့ HTML Code တွေ သာမက အခြား Programming တွေလေ့လာတဲ့အခါ Indent ကိုမှန်အောင်ရေးရပါ တယ်။

Indentation ဆိုတာသာလဲဆိုရင် “the spacing and alignment of lines of code to visually organize and structure a program” ပါ။ Spacing Alignment တွေကို ညီအောင် Visually Organize ဖြစ်အောင် Structure ကျအောင် ရေးနည်းပါဖြစ်ပါတယ်။



The screenshot shows a code editor with the following HTML code:

```

<head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Lwin Moe Paing | Profile</title>
</head>

```

Two numbered callouts point to specific parts of the code:

- 1** Points to the first two lines of the head section, which are aligned to the left.
- 2** Points to the third line of the head section, which is aligned to the right, demonstrating proper indentation.

ပုံထဲက Code တွေကို လေ့လာကြည့်ရအောင် <head> အဖွင့်နဲ့ </head> အပိတ်သည် တစ်တန်းတည်းရှိနေဖြီး၊ အတွင်းရှိ <meta> , <title> tag များကို “tab” spacing ခြားထားသည်ကို သတိပြုရှိကိုပါ။ ထိုသို့ ရေးထားသောကြောင့် Code ကို ကြည့်လိုက်တာနဲ့ ကျွန်ုတ်တို့က ဘယ် Element သည် ဘယ် Element ၏ အထဲတွင်တည်ရှိနေသည် ကို သိနိုင်ပါသည်။

အပြင်လောက Real Life Project များတွင် အခြားသော Developer များနဲ့ အတူတကွ ပူးပေါင်းရေးသားရသောကြောင့် Indent မှန်အောင် ရေးသားခြင်းသည် Code ပြန်လည် ပြုပြင်ခြင်း (maintaining) ဖော်ရရှုရလွယ်ကူခြင်း (readability) အတွက် များစွာ အထောက်အကူ ပြုပါသည်။



မကြာခင် NASA ကို Hack နိုင်မယ့်
HTML ကို လေ့လာတတ်မောက်ပါတော့မယ်
(HTML သည် Programming Language မဟုတ်ပါ)

HTML Elements

တကယ်က HTML က Hack လိုမပါဘူးနောက် တကယ်တွေထင်နေမှာစိုးလို့ပျော်။ ဒီအပိုင်းမှာတော့ User တွေအတွက် တွေ့ဖြင့်ရသော အပိုင်းများ (Visible Contents) အတွက် Element တွေကို စလေ့လာပါမယ်။

Heading Element များ <h1> to <h6>

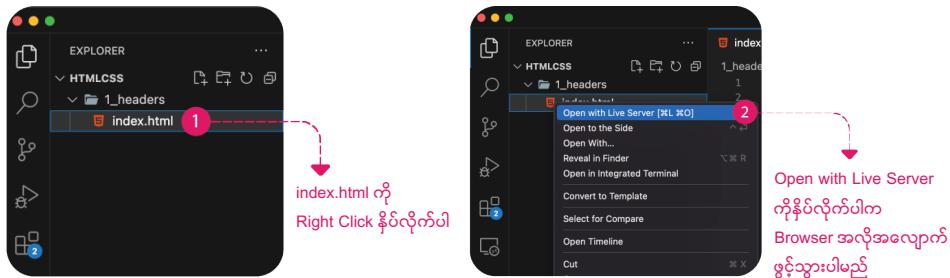
ခေါင်းစဉ်တွေအတွက် <h1> ကနေ <h6> Elements များကို သုံးပေးနိုင်ပါတယ်။ <h1> ကအမြင့်ဆုံး၊ အဓိကအကျဆုံး အတွက်အသုံးပြုပြီး ကျွန်ုတဲ့ ခေါင်းစဉ်တွေအတွက် Content ပေါ်မှာ မှတ်ည်းမြို့အပ်သလိုအသုံးပြုနိုင်ပါတယ်။

အရင်ဆုံး ကျွန်ုတဲ့ “1_headers” ဆိုပြီး Folder (ဖိုလ်ဒါ) လေးတည်ဆောက်လိုက်ပါမယ်။ အဲထဲမှာ “index.html” ဆိုတဲ့ file (Html ဖိုင်) တည်ဆောက်လိုက်ပါမယ် ပြီးရင်အောက်ကအတိုင်း ရေးသားပါမယ်။

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>ခေါင်းစဉ်</title>
  </head>
  <body>
    <h1>ခေါင်းစဉ် h1</h1>
    <h2>ခေါင်းစဉ် h2</h2>
    <h3>ခေါင်းစဉ် h3</h3>
    <h4>ခေါင်းစဉ် h4</h4>
    <h5>ခေါင်းစဉ် h5</h5>
    <h6>ခေါင်းစဉ် h6</h6>
  </body>
</html>
```

ရေးသားစဉ်တွင် <body> Element အတွင်းသာရေးသာခဲ့သည်ကို သတိပြုပေးပါ ထိုအတဲ့အတွက်တာ User မြင်ရတဲ့အပိုင်းများကို ရေးသာနိုင်မည်ဖြစ်ကြောင်း ပြောပြထားပြီးဖြစ်ပါတယ်။ ပြီးနောက်ကျွန်ုတဲ့ “1_headers/index.html” ဆိုသော ထိုဖိုင်ကို Browser ဖြင့် ဖွင့်လိုက်ပါမည်။

Browser ဖြင့် မဖွင့်တတ်ပါက ကျော်တော်တို့ Preparing For Coding မှာတူန်းက Extension များကို ထည့်ခဲ့သေးသည်ကိုအမှတ်ရပါ။ အဲထဲက Live Server Extension ကိုအသုံးပြုပြီး အောက်က ပုံထဲကအတိုင်း Live ServerRun နိုင်ပါတယ်။ သူက ကျော်တော်တို့ Save မှတ်လိုက်သည့်အပါ Browser တွင် Refresh ပြန်လည်လုပ်စရာ မလိုပဲ အလိုအလေ့ပျက် (auto) Refresh လုပ်ပေးသဖြင့် Code ရေးနေစဉ်အတွင်း အထောက်အကြပ်ပေးလိမ့်မယ်။



အထက်က အတိုင်းပြုလုပ်ရာတွင် Live Server ဖြင့် ဖွင့်ရန် အခက်အခဲဖြစ်နေပါက အောက်က ပုံအတိုင်း Folder ရှိရာ နေရာ (Directory) ကိုသွားပြီး တိုက်ရှိက် ဖွင့်နိုင်ပါတယ်။



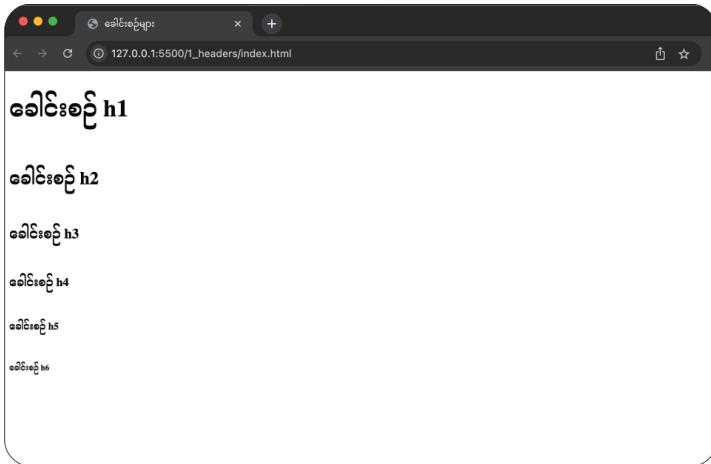
1

index.html ကို
Right Click နိုင်လိုက်ပြီး
Reveal in Finder ကိုနိုင်ပါ

2

index.html ဖိုင်ရှိသော Folder ထဲကိုရောက်လျှင်
Right Click နှုပ်ကာ Open With တို့ရွေးပါ
ပြီးနောက် Google Chrome (ဘို့) Browser
တစ်ခုစုရွေးပြီး ဖွင့်နိုင်ပါသည်

အညွှန်း (၁) အရ Mac တွင် Reveal In Finder ဖြစ်နိုင်ပြီး Windows တွင် Open in Folder ဆိုပြီးဖြစ်နိုင်ပါသည်။



ဒီလို Browser မှာ ပေါ်လာမှာကို မြင်ရမှာပါ။

Congratulation ပါဘု ပထမဆုံးအနေနဲ့ ခေါင်းစဉ်တွေအားလုံးကို ကျွန်ုတ်တို့ ရေးလိုက်နိုင်ပါပြီ။ ကျွန်ုတ်ဒီနေရာမှာ h1 ကနေ h6 အထိ Font Size ဘယ်လောက်ရှိတယ် ဘယ်လိုလိုဆိတာ ရေးသားမနေတော့ပါဘူး Browser က Default အနေနဲ့သတ်မှတ်ပြီးသား ရှိပေါ်မယ့်လည်း တကယ်လက်တွေ။ Project တွေမှာ CSS ဖြင့်သာသတ်မှတ်ရေးသားကြရတဲ့အတွက် ဖြစ်ပါတယ်။

Heading အတွက် Code တွေကို အောက် CodePen မှာသွားစမ်းကြည့်လို့ရပါတယ်
<https://codepen.io/lwinmoepaing/pen/ExOZQxb>

Code တွေကို Gist အနေနဲ့ တိုက်ရှိက်သွားကြည့်ချင်တယ် ဒေါင်းချင်တယ်ဆိုရင် အောက်ကလုံးကနေ သွားနိုင်ပါတယ်

<https://gist.github.com/lwinmoepaing/4fae35bf55b3cd2ce11d0663621d9b5d>

ဆက်ပြီး အဲြား Element များကို လေ့လာကြည့်ရအောင် ...

Paragraph စာပိုဒ်များအတွက် <p>

စာပိုဒ်များအတွက်ဆိုရင် ကျွန်ုတ်တို့ <p> Element ကိုအသုံးပြုပါတယ် သူက Paragraph ရဲ့ အတိကောက် p ဖြစ်ပါတယ်။ အရင်ဆုံး ကျွန်ုတ် “2_paragraph” ဆိုပြီး Folder (ဖိုလ်ဒါ) လေးတည်ဆောက်လိုက်ပါမယ်။ အဲထဲမှာ “index.html” ဆိုတဲ့ file (HTML ဖိုင်) တည်ဆောက်လိုက်ပါမယ် ပြီးရင် နောက်တစ်မျက်နှာရဲ့ ပုံတဲ့ကအတိုင်း ရေးသားလိုက်ပါမယ်။

```

2_paragraph/index.html

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>စာပိုဒ်များ | Lwin Moe Paing</title>
</head>
<body>
    <p>
        စာပိုဒ် (၁) အခရေးသားထားသော စာပိုဒ် ၁ အကြောင်းအရာသည် Paragraph ဖို့သော  

        တကာလုံးမှ p ကို ယူထားသော tag အကြောင်းအရာဖြစ်ပါသည်။
    </p>
    <p>
        စာပိုဒ် (၂) အရ အခရေးသားထားသော စာပိုဒ်သည် နေဂံတစ်ပိုဒ်ကို  

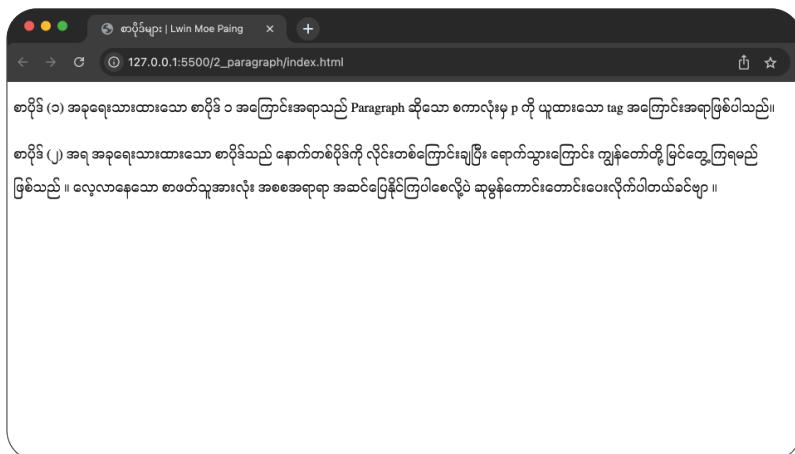
        လိုင်းတစ်ခုကြောင်းပြုပါ၍ ရောက်သွားကြောင်း ကျိုးကျင်တဲ့  

        ပြင်တွေ့ကြော်ပြုသွားပါ။ လေလာမျေားသော စာဖတ်သာအေးလုံး အစစအရာရာ  

        အဆင်ပြန်ကြပါလေလုံးပါ၍ ဆုမ္ပန်ကောင်းတောင်းပေးလိုက်ပါတယ်ခင်ဗျာ။
    </p>
</body>
</html>

```

<p> Element ကိုရေးရာတွင်လည်း <body> Element အတွင်းသာရေးသာခဲ့သည်ကို
 ထပ်သတိပြုပေးပါ။ စာဖတ်သူတို့က <p> Element အတွင်း ကိုယ်ကြိုက်နှစ်သာက်ရာ
 စာသားကို ထည့်သွေးအသုံးပြု ရေးသားနိုင်ပါသည်။ အခုအပေါ်ကအတိုင်း ကျွန်ုတ်ကော်
 က ရေးသားပြီး "2_paragraph/index.html" ဖို့သော ထို့ဖိုင်ကို Browser ဖွင့်ဖွင့်လိုက်
 သောအခါ အောက်က ပုံအတိုင်း...



Browser မှာ ပေါ်လာမှာကို မြင်ရမှာပါ။ Congratulation ပါ၍ နောက်ထပ် Paragraph Element ကိုရေးသားလိုက်နိုင်ပါပြီ။ Code Pen နဲ့ Git's Gist ကို နောက်တစ်မျက်နှာမှာ
 ထည့်ပေးထားပါသည်။

Paragraph အတွက် CodePen မှာသွားစမ်းကြည့်လိုပါတယ်

<https://codepen.io/lwinmoeapaing/pen/KKraBOE>

Code တွေကို တိုက်ရိုက်သွားကြည့်ချင်တယ် ဒေါင်းချင်တယ်ဆိုရင်
အောက်ကလင့်ကနဲ သွားနိုင်ပါတယ်

<https://gist.github.com/lwinmoeapaing/83b4f774e107894fb9c803275912d7>

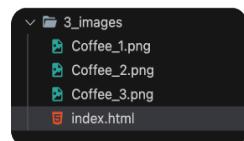
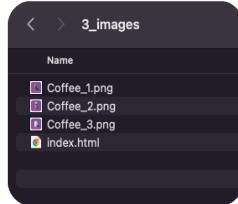
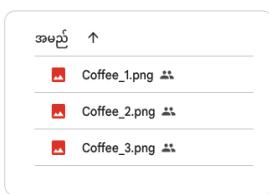
နောက်ထပ် ကျွန်တော်တို့ Image Element ကို လေ့လာကြည့်ကြမယ်

Image များ ပုံများအတွက်

အရင်ဆုံး ကျွန်တော် “3_images” ဆိုပြီး Folder (ဖိုလ်ခါ) လေးတည်ဆောက်လိုက်ပါ မယ်။ အဲထဲမှာ “index.html” ဆိုတဲ့ file (HTMLဖိုင်) တည်ဆောက်လိုက်ပါမယ် ပြီးရင်။

အောက်တွင် ပေးထားတဲ့ Google Drive Links တွေကိုဝင်ပြီး Coffee ပုံလေးတွေကို ဒေါင်းလေ့ဆွဲလိုက်ပါ။

https://drive.google.com/drive/folders/1ljuSuA7Y9acnW7Rof82LPKcGy7-AYz_?usp=drive_link



1

Google Drive ထဲရှိ
Coffee_1, Coffee_2, Coffee_3
ဆိုတော့ ကော်ပိ Png ဖိုင်(ရ) ပိုင်လုံး
Download လွှာပေးပါ

2

ပြုလုပ်ထားသော 3_images
Folder (ဖိုလ်ခါ) အတွင်းသို့
ထို့ ၃ ပုံလုံး Copy ကူးယူလိုက်ပါမည်

3

VS code အတွင်း
ထို့တဲ့ဆို ဖြင့်ရပါတော်မည်

ပြီးနောက် နောက်တစ်မျက်နှာတွင် Element ကိုအသုံးပြုပြီး ကျွန်တော်တို့ Coffee ပုံလေးတွေကို Browser ပေါ်မှာ ပုံဖော်ကြည့်ရအောင်

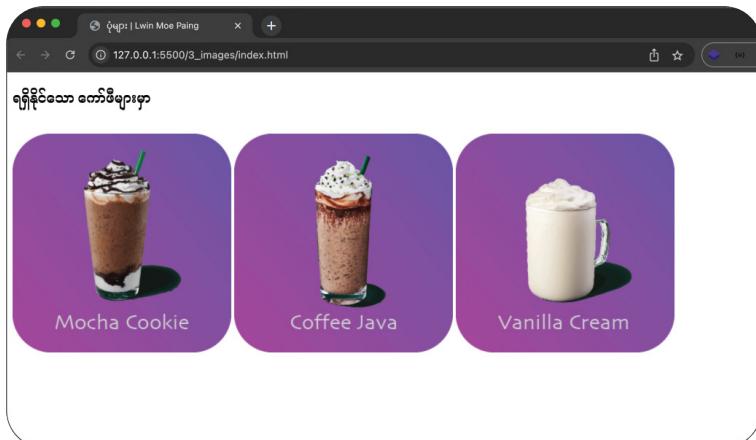
```
3_images/index.html

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<title>ဗျား | Lwin Moe Paing</title>
</head>
<body>
<h3>မျိန်ငြေသာ ကော်ဖူးမှာ</h3>



</body>
</html>
```

 Element တွင် ကျွန်တော်တို့ ဖော်ပြီးခဲ့သော Attributes များကို သုံးထားသည်ကို တွေ့ရပါတယ်။ သူမှာ Attribute Name ဖြစ်သော "src" သည် ပုံရဲ့ Path လမ်းကြောင်း (ဘို့မဟုတ်) URL ကိုထည့်သွင်းပြီး ရေးသားနိုင်တယ်။ "src" ဆိုမှာ Source အတိကောက် ဖြစ်ပြီး ပုံရဲ့အရင်းအမြစ်ကို ဆိုလိုခြင်းလဲဖြစ်တယ်။ Element ရဲ့ ထူးဌားချက်မှာ အဖွင့် Tag (Opening) ပဲရှိပြီး အပိတ် Tag (Closing Tag) မရှိဘူး သို့ ပေါ့မယ် နောက်ဆုံးမှာ ">" နဲ့ဆုံးတာကို သတိပြုစေချင်ပါတယ် (Self Closing Tags) လိုလည်းခေါ်ပါတယ်။



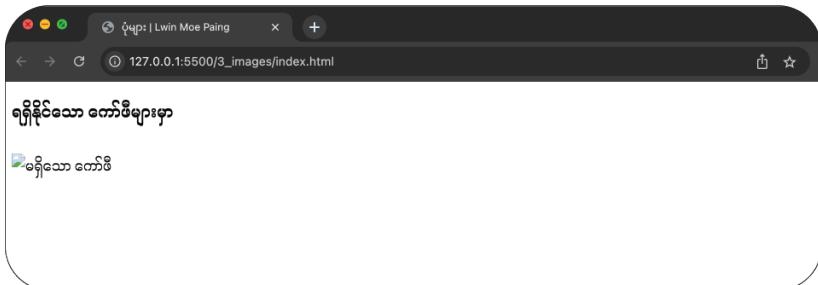
ယခုကဲ့သို့ Browser တွင် ပေါ်သွားမည်ဖြစ်သည်။ သို့ပေါ့မယ် တခါတရုံး ကျွန်တော် တို့ ပေးလိုက်သော Source "src" ပါတ်လမ်းကြောင်းမှာ မှားယွင်းစွာ ထည့်မြှင်းများ (ဘို့မဟုတ်) အမှန်တကယ်မရှိသော ပုံရဲ့ တည်နေရာကို ထည့်မိတတ်သည်။

ကျွန်ုတ်တို့ ထဲက Attribute src ထဲတွင်မရှိသော ကော်ဖို့လေးတစ်ခုကို ခေါ်ယူထည့်လိုက်ပါမယ်။ "Coffee_3.png" အထိပဲ ကျွန်ုတ်တို့မှာရှိတာပါ မရှိတဲ့ ပုံတစ်ခုကို ကျွန်ုတ်တို့ "Coffee_4.png" အဖြစ် src ထဲတွင်ထည့်လိုက်ပြီး အောက်ကပုံ အတိုင်း ရေးလိုက်ပါမယ်။

```
<body>
<h3>မရှိနိုင်သော ကော်ဖို့များ</h3>

</body>
```

အောက်က ပုံထဲကလို့ မရှိနိုင်သော ကော်ဖို့များ ပြနေမှာဖြစ်ပါတယ် ဘာလို့ဆို ကျွန်ုတ်တို့ Code ကို ပြန်ကြည့်တဲ့အခါ Attribute "alt" ကို တွေ့ရပါမယ် သူရဲ့ value အဖြစ် စာသား "မရှိနိုင်သော ကော်ဖို့" ဆိုပြီးထည့်ထားတဲ့အတွက်ကြောင့် ပုံကို မတွေ့တဲ့အခါ ပြပေးသွားတာဖြစ်ပါတယ်။ Alternative Information ခဲ့ အတိုကောက် အရေးက "alt" ယူထားတာဖြစ်ပါတယ် အကယ်လိုင်းမကောင်းလို့ ပုံကိုမရတဲ့အခါမျိုးမှာ (သို့) မရှိတဲ့ ပုံကိုလုပ်ဖို့မှာ Browser က ပုံကို ဘာသာမပြန်ပေါ်နိုင်တဲ့အတွက် alt ထဲက စာသားကို ဖော်ပြလိုက်တာဖြစ်ပါတယ်။



ဒါဆို "title" Attribute က ကော ဘာလုပ်မဲ့ ထည့်ထားတာလည်းမေးစရာရှိပါတယ်။ ကျွန်ုတ်တို့ ပထမ ပုံထဲကအတိုင်း Coffee ခွက် (၉) ပြန်ရအောင် အရင်အတိုင်း Code တွေကို အမှန်ပြန်ပြင်လိုက်ကြမယ်။ Coffee_1.png, Coffee_2.png, Coffee_3.png ရယ်ကို Element နှဲအတူ ပထမအတိုင်းပုံတွေပေါ်အောင် ရေးလိုက်ကြမယ်၏။ အထဲမှာ "alt", "title" ရောကို ထည့်ဖို့ မပေါ်ပါနဲ့၏။ ပြီးရင် Browser မှာ ဖွင့်လိုက်ပြီး ကော်ဖို့ ပုံတွေအပေါ် Mouse အနေနဲ့ (၃) စလှန်လောက်ကြာအောင် မလုပ်ပဲထားကြည့်ပါ။



စာဖတ်သူတို့က title ရဲ့ Attribute value မှာ ကိုယ်ပြောင်းချင်တဲ့ စာသားလေးတွေ
ပြောင်းလဲ ပေးပြီး စမ်းသပ်ကြည့်လို့ရပါတယ်။

Images အတွက် Code တွေကို အောက် CodePen မှာသွားစမ်းကြည့်လို့ရပါတယ်
<https://codepen.io/lwinmoeapaing/pen/dyQNEQE>

Gist Code အတွက် အောက်ကလင့်ကနေ သွားနိုင်ပါတယ်

<https://gist.github.com/lwinmoeapaing/20e6c6471790f6940a77e6bdf1dea727>

List များဖော်ပြရာအတွက်

Browser ပေါ်တွင် List များကို ဖော်ပြရတွင် နှင့် ဆိုတဲ့ Element (J) မျိုးကို
အသုံးပြုနိုင်ပါသည်။ က UnOrdered List ရဲ့ အတိုကောက် ပါ အစီအစဉ်မကျတဲ့
list လို့ခေါ်ပါတယ်။ တကယ်တော့ List ရဲ့ အရေးမှာ နံပါတ်တွေအစဉ်နဲ့ဖော်ပြထား
တာကို နဲ့ အသုံးပြုပြီး Order List သဘောကိုဆိုလိုချင်တာပါ။

 ကတော့ List Item ရဲ့ အတိုကောက်ဖြစ်ပြီး ခုကာက နှင့် တို့ရဲ့ အတွင်း
ပိုင်းမှာ ရေးလေ့ရှိပါတယ်။

အရင်ဆုံး ကျွန်တော် “4_lists” ဆိုပြီး Folder (ဖိုလ်ဒါ) တည်ဆောက်လိုက်ပါမယ်။ အဲ
ထဲမှာ “index.html” ဆိုတဲ့ file (HTMLဖိုင်) တည်ဆောက်လိုက်ပါမယ် ပြီးရင် နောက်
တစ်မျက်နှာက အတိုင်း ကျွန်တော်တို့ Code ကြည့်ပါမယ်ဗျာ။

```

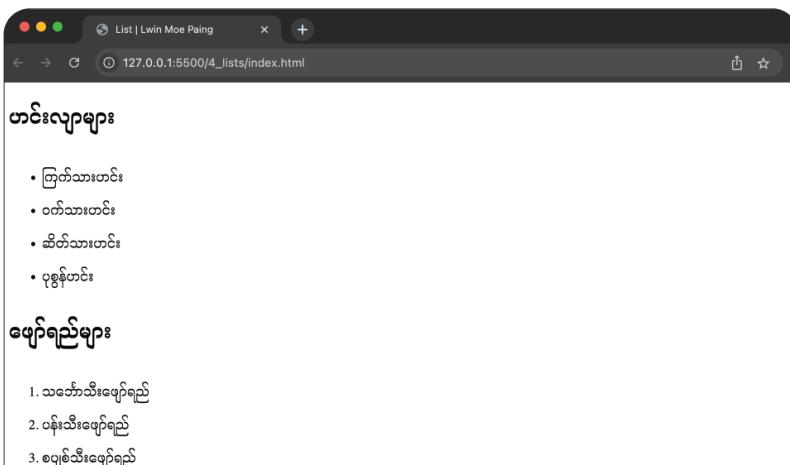
4_lists/index.html

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>List | Lwin Moe Paing</title>
</head>
<body>
    <h2>ယင်းအပြည့်</h2>
    <ul>
        <li>ကြော်သားယင်း</li>
        <li>ဝင်္ဂီသားယင်း</li>
        <li>ဆိတ်သားယင်း</li>
        <li>ပုစ္န်ယင်း</li>
    </ul>

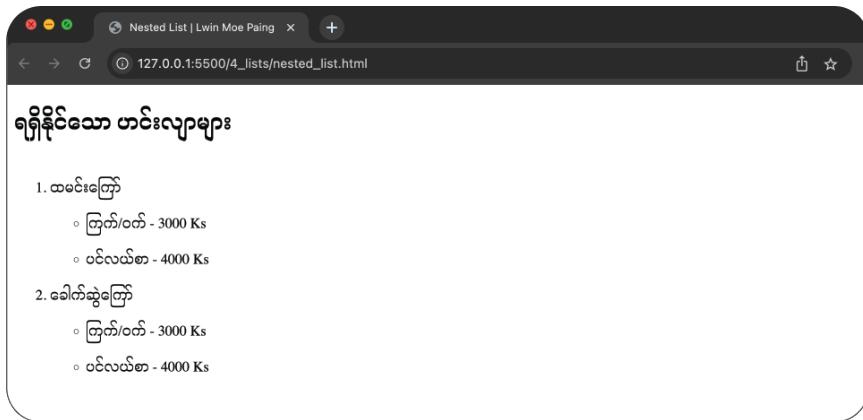
    <h2>မျှော်လုပ်များ</h2>
    <ol>
        <li>သဘောသီးမျှော်လုပ်</li>
        <li>ပန်းသီးမျှော်လုပ်</li>
        <li>စပါစ်သီးမျှော်လုပ်</li>
    </ol>
</body>
</html>

```

သတိထားရမည့်မှာ Element သည် (သို့) များရဲ့ အတွင်းမှာသာ ထားလေ့ရှိပါတယ် List Item ဆိုတဲ့အတိုင်း Ordered List (သို့) Unordered List ရဲ့ အတွင်းပိုင်း Item အဖြစ်တည်ရှိပါသည်။ Browser တွင် Run လိုက်ပါက အောက်ပါ ပုံအတိုင်းတွေရပါမည်။ Ordered List အတွက် Item များသည် နံပါတ်စဉ်အလိုက် ဖော်ပြန်ပြီး၊ Unordered List အတွက်တော့ Disc Style အနေဖြင့် စက်ပိုင်းအမည်းများဖြင့် ပုံဖော်သွားပါမည်။



"4_lists" Folder (ဖိုလ်ဒါ) ထဲမှာ နောက်ထပ် ဖိုင်အသစ်တရ "nested_list.html" ဆုံး
တဲ့ file (HTMLဖိုင်) တည်ဆောက်လိုက်ပါမယ်။ အောက်က Browser မှာ ပေါ်တဲ့ ဗုံး
အတိုင်း ကျွန်ုတ်တို့ List တွေသည် သာမန်မဟုတ်ဘဲ အတွင်းထပ်ဆင့် အသုံးပြုခြင်း
များ ရှိနိုင်ကြောင်းလေ့လာကြရအောင် ...



ပုံကိုကြည့်လိုက်ပါ ပထမဆုံး မြင်ရမယ့် List က နံပါတ်စဉ်နဲ့ပါ အဲဒီအတွက် ပထမဆုံး Order List ဖြစ်မှာပါ Element ကိုအသုံးပြုပါမယ် နောက် သူ့အထဲမှာ အမြဲ List Item တွေရှိပါတယ် ထမင်းကြောက်ကို ထည့်လိုက်ပါမယ် သူ့ရဲ့အောက်မှာတော့ နောက်ထပ် List တစ်ခုထပ်ရှိပါတယ်။ ဟင်းလျာရဲ့ အမျိုးအစားနဲ့ သျေနှုန်းကို ထည့်ပေးရမှာပါ အဲအတွက် ကျွန်ုတ်တို့ List တစ်ခုထပ်တည်ဆောက်ရပါမယ်။

သို့ပေမယ့် သေချာကြည့်တဲ့အခါ နံပါတ်စဉ်နဲ့ List မဟုတ်ပါဘူးအဲအတွက် Unordered List ဖြစ်တဲ့ Element ကို အသုံးပြုရပါမယ်။ ပြောခဲ့တဲ့အတိုင်း တို့ အတွင်းမှာ List Item ဖြစ်တဲ့ Element ကို အသုံးပြုရပါမယ်
ကြောက်/ဝက် - 3000 Ks
ပင်လယ်စာ - 4000 Ks

နှစ်မျိုးစလုံးကို နဲ့ ရေးလိုက်ပြီးရင် အောက်က ခေါက်ဆွဲကြောက်လည်း ထိနည်း
လည်းကောင်း ဆက်ရေးသွားလို့ရပါပြီ။ အခုကျွန်ုတ်တော်ရှင်းပြတာကို မြင်သွားအောင်
တမျက်နှာ နောက်တစ်မျက်နှာမှာ Code ရှင်းလင်းချက် ပြပေးထားပါတယ်။



The screenshot shows a mobile browser window with three dots at the top left. The title bar says "4_lists/nested_list.html". The page content is as follows:

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Nested List | Lwin Moe Paing</title>
  </head>
  <body>
    <h2>ရှိခိုင်သော ဟင်းလျှပ်ပါ</h2>
    <ol>
      <li>
        ထမင်းကြော်
        <ul>
          <li>ကြော်/ဝင် - 3000 Ks</li>
          <li>ပွဲလယ်စာ - 4000 Ks</li>
        </ul>
      </li>
      <li>
        ငါးရှိခိုင်းကြော်
        <ul>
          <li>ကြော်/ဝင် - 3000 Ks</li>
          <li>ပွဲလယ်စာ - 4000 Ks</li>
        </ul>
      </li>
    </ol>
  </body>
</html>

```

Code တွေဖတ်ကြည့်လိုက်တဲ့အခါ ပိုရှင်းလင်းသွားမယ်လို့ မျှော်လင့်ပါတယ်။

ထဲမ Ordered List နဲ့ Unordered List အတွက်ကို

အောက်က Codepen မှာစမ်းလို့ရပါတယ်

<https://codepen.io/lwinmoepaing/pen/QWJpbdN>

Gist Code ကို ဒီကနေသွားနိုင်ပါတယ်

<https://gist.github.com/lwinmoepaing/9329f17a199b43ce836cce2de24410e1>

အတွင်းသက် တစ်ကြိမ်ထပ်ရှိသော List ကိုစမ်းရန်အတွက် အောက်က Codepen

ကနေတစ်ဆင့် သွားရောက် စမ်းကြည့်နိုင်ပါတယ်

<https://codepen.io/lwinmoepaing/pen/RwqpPKd>

Gist Code ကတေသာ အောက်က link လင်နိုင်ပါ

<https://gist.github.com/lwinmoepaing/b3aa4c14a24592356f71bfcef23828f>

Anchor Link လင့်ခဲများအတွက် <a>

HTML link တွေဟာ Hyperlinks တွေဖြစ်ကြပါတယ်။ အဲြား Document (သို့) အဲြား Page တွေကို ခုန်ကူးနိုင်ပါတယ်။ အဲြားလို ဆောင်ရွက်ဖို့အတွက် <a> Element ကို အသုံးပြုရပါတယ်။ <a> က Anchor ရဲ့အတိုကောက်ကို ယူထားတာဖြစ်ပြီး သူ ကတော့ Link တွေချိတ်ဆက်ပေးနိုင်တဲ့အပ် ကိုယ့်ရဲ့ လက်ရှိ Page အတွင်း အဲြား Element တွေကိုလည်း ခုန်ကူးပေးနိုင်ပါတယ်။ ဖုန်းကိုခေါ်တဲ့အပါ (သို့) Mail ပို့တဲ့အပါ မျိုးမှာလည်း အသုံးချုနိုင်ပါသေးတယ်။

အရင်ဆုံး ကျွန်ုတ်တို့ “5_links” ဆိုတဲ့ Folder (ဖိုလ်ဒါ) အသစ်တစ်ခုတည်ဆောက် လိုက်ပါမယ်။ ဖိုင် (၂) ခုကိုတည်ဆောက်လိုက်ပါမယ်

ပထမတစ်ခုအတွက် “index.html” ရယ် ပြီးတော့ ဒုတိယ page အတွက် “second.html” ဆိုပြီး တည်ဆောက်လိုက်ရအောင်။

index.html အတွင်းအောက်က Gist Code ထဲကအတိုင်းကူးယူရေးသားလိုက်ပါ။

<https://gist.github.com/lwinmoeapaing/25dfb8e239575f1b488f1a0e849e0fc9>



```

5_links/index.html

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Links | Lwin Moe Paing</title>
</head>
<body>
    <h2 id="top_h2">Links တွေအတွက် Home Page ငော် ဖြန့်ပါတယ်</h2>
    <h3>Video တွေကြည့်နိုင်ခဲ့</h3>

    <a href="http://youtube.com">Youtube ကိုသုံးမယ် </a>
    <br />
    <a href="https://tiktok.com" target="_blank">TikTok ကိုသုံးမယ် </a>
    <br />

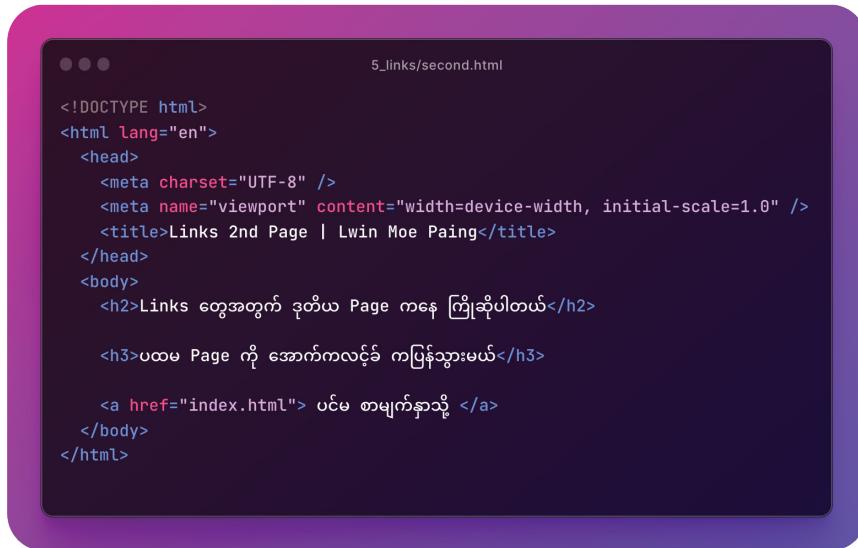
    <h4>Phone လည်းခေါ်စေတယ်</h4>
    <a href="tel:+959420059241"> ကိုစွမ်းပိုင်ကို ဖုန်းခေါ်မယ် </a>

    <p id="paragraph_1">စာတို့ (၁) - အစေးရောထားသည့် Testing စာတို့ပြုစာညွှေ့</p>
    <p id="paragraph_2">စာပို့ (၂) - အစေးရောထားသည့် Testing စာပို့ပြုစာညွှေ့</p>
    <p id="paragraph_3">စာပို့ (၃) - အစေးရောထားသည့် Testing စာပို့ပြုစာညွှေ့</p>
    <hr />
    <a href="#paragraph_7">လင့်ခဲ - အောက်ဖူးစီစာပို့အဲ့ ခုန်းမျိုး...</a>

    <p id="paragraph_4">စာတို့ (၄) - အစေးရောထားသည့် Testing စာတို့ပြုစာညွှေ့</p>
    <p id="paragraph_5">စာပို့ (၅) - အစေးရောထားသည့် Testing စာပို့ပြုစာညွှေ့</p>
    <p id="paragraph_6">စာတို့ (၆) - အစေးရောထားသည့် Testing စာတို့ပြုစာညွှေ့</p>
    <p id="paragraph_7">စာပို့ (၇) - အစေးရောထားသည့် Testing စာပို့ပြုစာညွှေ့</p>
    <a href="#">Page ထံပေးအောင် [ပြောက်ရနိုင်] ...</a>
    <br />
    <a href="second.html">ဒုတိယ တမ်းကြည့်နိုင် ...</a>
</body>
</html>

```

second.html အတွင်းအောက်က Gist Code ထဲကအတိုင်းကူးယူရေးသားလိုက်ပါ။
<https://gist.github.com/lwinmoepaing/bf6496930dc10e749bfd20e3f8be596b>

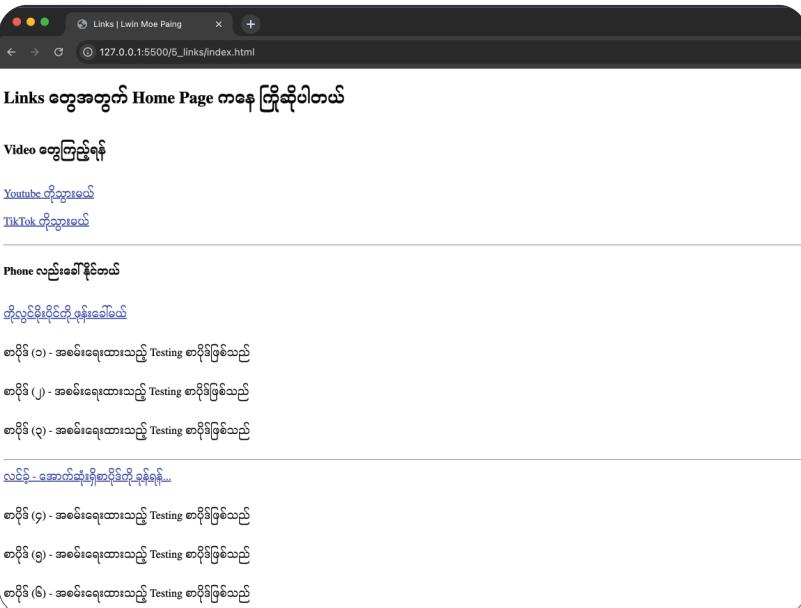


```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Links 2nd Page | Lwin Moe Paing</title>
  </head>
  <body>
    <h2>Links တွေအတွက် ဒုတိယ Page ကနေ ကြိခိုပ်ပါတယ်</h2>

    <h3>ပထမ Page ကို အောက်ကလုံး ကပြန်သွားမယ်</h3>

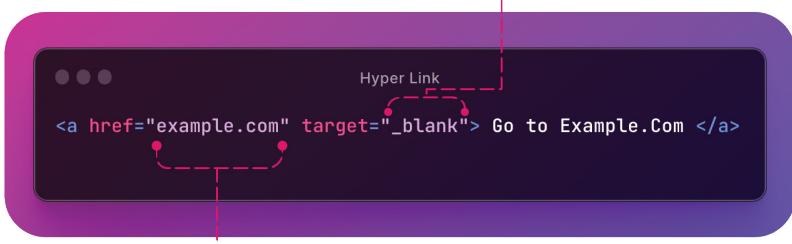
    <a href="index.html"> ပင်မ စာမျက်နှာသို့ </a>
  </body>
</html>
```

ပထမ ပြုလုပ်ခဲ့တဲ့ index.html ကို Browser ပေါ်တွင် ဖွင့်ကြည့်လိုက်ပါ။ အောက်က အတိုင်း ကျွန်တော်တို့ ဖြင့်ရပါတယ်မယ်။ ပြီးရင် အကုန် Links တွေကို လိုက်စမ်းသပ် ကြည့်နိုင်ပါတယ်။



ကျွန်တော်တို့ စမ်းပြီးသွားတဲ့အချင် အောက်က ပုံလေးကို အရင် လေ့လာကြည့်ရအောင်

Link ကိုခေါ်လိုက်ရင် Browser Target ကို
ရွှေးပေးရတာပါ။ Default အနေနဲ့က “_self” ပါ ကိုယ့်လက်ရှိ
Document ဖူးပဲ ၏ခေါ်မယ်ဆိုတဲ့ သဘောပါ။ _blank ဆိုရင်
နောက်ထက် Tab တစ်ခုထပ်ခေါ်သွားပါမယ်။



Attribute “href” အတွက် Value ကိုတော့
URL သို့မဟုတ် စိမ့် Page ရဲ့ Path လမ်းကြောင်းကိုပေးပြီး
ချိတ်ဆက်နိုင်ပါတယ်။ ဒါအပြင် “mailto:some@mail.com”
ဖြင့်လည်း Email တွေအတွက်အသုံးပြန်ပြီး Phone နဲ့ပါတ်
တွေခေါ်ခြင်ရန် အရေ့က prefix(tel) တို့သုံးပြီး
“tel:+959420000000” အနေနဲ့ ရေးနိုင်ပါတယ်

အခု Code တွေကိုလေ့လာမိတဲ့အချင်မှာ စာဖတ်သူတို့အတွက် Element အသစ် (J) ခု တိုးလာတာကို သတိပြုမိကြပါသလားခင်ဗျာ။ ပထမတစ်ခုက
 ဆိုတဲ့ Element ဖြစ်ပြီး နောက်တစ်ခုကတော့ <hr /> element လေးဖြစ်ပါတယ်။

 Element ကတော့ Line Break ဆိုတဲ့ အဓိပ္ပာယ်ပါ သူက နောက်တစ်လိုင်းကို
ဆင်းပေးတဲ့အလုပ်ကိုလုပ်ပါတယ်။ သတိထားမိမလားမသိဘူး <h1> , <p> အစရှိတဲ့
Element တွေကို ရေးခဲ့တုန်းကို အောက်ကို Auto ကျသွားပါတယ် သူတို့က Parent
Container မှာရှိသလောက် အကျယ် (Width) ကိုယူသွားတာပါ Block Element လို့
လည်းခေါ်ပါတယ်။

<a> , တို့ကိုရေးခဲ့တုန်းကတော့ အောက်ကို တစ်လိုင်းမဆင်သွားပါဘူး ဘေး
ကပ်ပုံမှာပဲ နောက်ထပ် တစ်ခုတိုးလာတာကို သတိထားမိကြမယ်ထင်တယ်။ သူတို့
ကျတော့ Inline Element လို့လည်းခေါ်ပါတယ် သူ့ကိုယ်ပိုင်ရှိသလောက် အကျယ်ကိုပဲ
ယူသွားတာပါ။ အဲအတွက် ကျွန်တော်တို့က နောက်တစ်လိုင်းဆင်ချင်တဲ့အခါ

ဆိုတဲ့ Element ကိုသုံးပြီး ဆင်းလို့ရပါတယ်။ <hr /> ရယ့် အရှည်ကတော့ Horizontal
Rule လို့ခေါ်ပါတယ် အလျေားလိုက် ရောဖြင့်ညီ မျဉ်းကိုစွဲတဲ့အခါမှာ သုံးပါတယ်။

နောက်တစ်ခုကတော့ Document ထဲမှာပဲရှိတဲ့ Element အချင်းချင်းကို Link လုပ်တာပဲ ဖြစ်ပါတယ်။ အဲလိုမျိုး လုပ်ဆောင်ရန် အချက် (J) ချက်လိုပါတယ်။

ပထမအချက်က ကိုယ် Link လုပ်ချင်တဲ့ Element ကို ID (Identifier) Attribute ပေးလိုက် ရပါမယ်။ ပြီးတော့ Unique Value ဖြစ်ရပါမယ်။ ဆိုလိုချင်တာက ကိုယ် ပေးလိုက်တဲ့ ID Value က တစ်ခုပဲရှိရမှာပါ အခြား Element တွေက အဲ Value တန်ဖိုး သုံးထားလို့မရပါဘူး။

ဥပမာ ကိုယ်က <h1> Element ကို <h1 id="header"> Header </h1> လိုပေးလိုက်ပါ။

ဒုတိယအချက်အနေနဲ့ အဲဒီ Header ကို Link လုပ်မယ့် <a> Element ကို ရေးပေးရပါ မယ်။ ကိုယ်က ခုကဗောပေးထားတဲ့ ID value ကို အရော့က "#" sign လေးခံပြီး href (Attribute) ထဲ ထည့်ပြီးချိတ်လိုက်ရှုပါပဲ။

ဒီလိုပါ - Link to Header

ရေးသားလိုက်ပါက ခုကဗော #header ဆိုတာ <h1> Element ရဲ့ ID တန်ဖိုးကို ချိတ် လိုက်ခြင်းဖြစ်ပါတယ်။ ဒါဆိုရင် Document တစ်ခုတည်းမှာရှိတဲ့ Element အချင်းချင်း ချိတ်နိုင်သွားပါပြီ။

သူက ဘယ်နောရာအသုံးဝင်လည်းဆိုရင် ကျွန်တော်တို့ ပြေချင်တဲ့ စာသားက Page ရဲ့ အောက်ဆုံးမှာဖြစ်နေတယ်ဆိုပါဆို အားဆုံး လင့်ခိုင်လုပ်လိုက်ပြီးတပါတည်း တန်းချိတ် ပေးလိုက်တာဖျိုးပါ။ User က Mouse နဲ့ အောက်ကို အကြောင်း Scroll ခွဲကြည့်စရာမ လိုတော့ပါဘူး (သို့) အောက်ဆုံးကနေ အပေါ်ဆုံး ကိုပြန်ရောက်ချင်တဲ့ အခါမျိုးမှာလည်း အပေါ်ဆုံးက Element ကို လင်ခဲ့လိုက်ပြီး မြန်မြန်ဆန်ဆန် သွားနိုင်ပါတယ်

အောက်က ပုံလေးကို တစ်ချက်ကြည့်ပေးပါ။



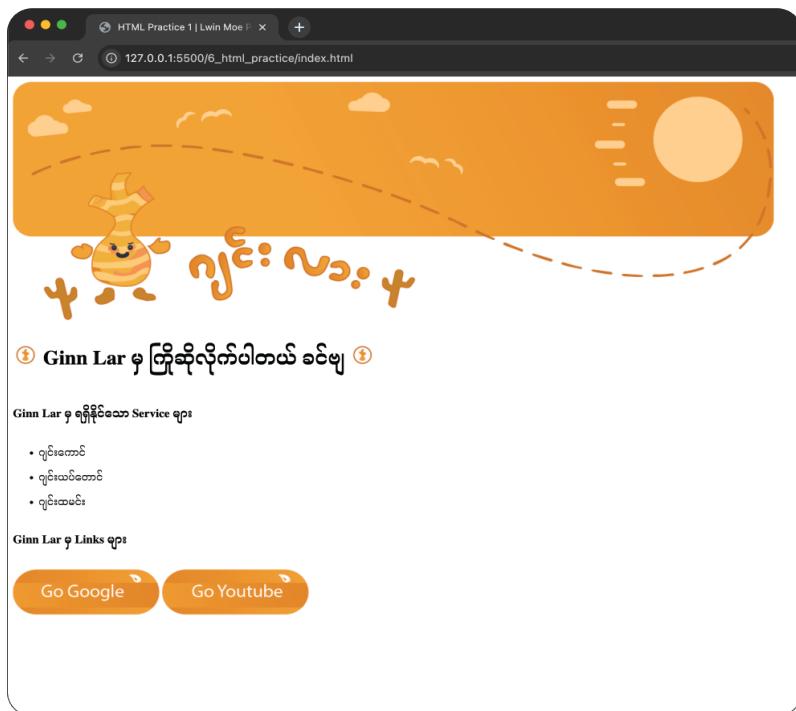
#top_h2 ဆိုသည့်မှာ အပေါ်က Element ရဲ့ Id value ကို ထုတ်ထားခြင်းဖြစ်သည်

ရေးသားခဲ့သော Anchor Link နှင့်ပိုင်ရာ Code တွေကို
အောက်က CodeSandBox လည့်ကန် သွားရောက် စမ်းနိုင်ပါသည်။
<https://codesandbox.io/s/5-links-y5wndm>

From Theory to Practice: Your First HTML

အချက်တော့ ကျွန်ုတ်တို့ Practice လုပ်ဖို့အချိန်ကိုရောက်ရှိလို့လာဖြိုဖြစ်ပါတယ်။
Heading Elements တွေဖြစ်တဲ့ <h1> ... <h6> အတိ । Paragraph <p> । Image
 । Lists । Links <a> တွေကို အမျိုးမျိုးလေ့လာခဲ့ပြီး ဖြစ်ပါတယ်။
ကျွန်ုတ်က Folder ကို “6_html_practice” လို့ပဲ ဖိုလ်ဒါ နာမည်လေးပေးပို့ကိုမယ် ဗျာ။
ကျွန်ုတ်တို့ အောက်က Website ပုံလေးကို ပေးထားပါမယ်။ အဲ့ပုံလေးအတိုင်းထွက်
အောင် စာဖတ်သူတို့က ရေးပေးရမှာပါ။ အဲ့ပုံထဲက လိုအပ်သော Image File တွေကို
အောက်က Google Drive Link က နေ Assets တွေကို Download သွားဆဲပေးပါ။

https://drive.google.com/drive/folders/1eM7jiS6EF4J9UhDk822PRtF80S-yFpe-?usp=drive_link



အဖြောက်တော့ နောက်တစ်မျက်နှာမှာ ဖော်ပြပေးထားမှာပါ သို့ပေါ့မယ် တတ်နိုင်သမျှ
ကိုယ့်ဘာသာ အဖြော်ရေးနိုင်ဖို့မျှော်လင့်ပါတယ်ခင်ဗျာ။ ရေးသားနေစဉ်မှာ မှားသွား
မှာ တို့ ကိုယ့်ရဲ့ Code က Quality မကောင်းတာတို့ကို စိတ်ထဲမထည့်ပါနဲ့ အရင်ဆုံး
Result ကိုသာ ထွက်အောင် အရင်လုပ်စေချင်ပါတယ်။

Code ကို အောက်က Gist မှာသွားရောက် တိုက်ကြည့်ပြီးလေ့လာနိုင်ပါတယ်။ ကုလ်ပိုင်ပုံစံဖြင့် Result အရ ကောင်းမွန်စွာထွက်နေဖြီး ဖြစ်သောလည်း ကျွန်ုတော် ပေးထားသော Code နဲ့တူညီမှ မှန်တယ်လို့မဆိုလိုပါ။ Element တွေကို မှန်ကန်စွာ အသုံးပြထားနိုင်လျှင်ပဲ အဆင်ပြေနေဖြင့်ဖြူးပြု။

<https://gist.github.com/lwinmoepaing/9a91d954a9e41953d90bc94a0460d15d>

အောက်က CodePen မှာလည်း Online ပေါ်မှာ တိုက်ရှိက် Run ကြည့်နိုင်ပါသည်။

<https://codepen.io/lwinmoepaing/pen/zYMZMdy>

လေ့လာကြည့်တဲ့အခါမှာ Nested Element အဖြစ်တွေ့ကို ကျွန်ုတော်တို့ တွေ့ခဲ့ရမှာ ဖြစ်ပါတယ် အပြင်လက်တွေ့ Project တွေမှာ Element တစ်ခုရဲ့အတွင်း အခြားသော Elements များစွာ ပါဝင်နေကြမှာပါ။

အခု အပေါ်က Code တွေထဲမှာ Link တွေကို Image Element နဲ့ ချိတ်ဆက်အသုံးပြုသွားသည်ကို မြင်တွေ့ရမှာပါ။ ပုံတွေအတွက် ကိုအသုံးပြုခဲ့ပေါ်မယ် တကယ်တန်း သူ့ရဲ့ အပေါ်ကနေ <a> Element ကို Wrap အပ်ပြီး အသုံးပြုခဲ့တာကို မြင်တွေ့ခဲ့ရမှာပါ။ အလားတွေ <h2> Heading Element ထဲမှာလည်း ကျွန်ုတော်တို့လိုချင်တဲ့ Icon လေးတွေပေါ်ဖို့အတွက် တွေထည့်သွင်းရေးသားခဲ့သည်ကိုလည်း သတိပြုမယ်လို့ ထင်ပါတယ်။ အောက်က ပုံကိုလေ့လာကြည့်ရအောင် အတွင်းသာက် အရ ပုံကို Browser ကဖော်ပြသွားပေါ်မယ့် သူ့ကို အုပ်ထားတဲ့ <a> Anchor Element ကြောင့် အုပ်ပုံကို နိုင်လိုက်ရင် သက်ဆိုင်ရာ Link ကို ဆက်သွယ်သွားပေးမှာပါ။

● ● ●

Nested Link With Image

```
<a href="https://www.google.com">
    
</a>

<a href="https://www.youtube.com">
    
</a>
```

Table အသာဓမ္မားတည်ဆောက်ခြင်း

ဒီအပိုင်းမှာ အသာဓမ္မားတည်ဆောက်ခြင်း `<table>` Element ကိုအသုံးပြုတာကို လေ့လာသွားမှာဖြစ်ပါတယ်။ `<tr>` Element က Table Row ရဲ့အတိကောက်ကိုယူထားတာဖြစ်ပြီး သူက Row တစ်ခုတစ်ဆောက်တဲ့အခါမှာအသုံးပြုပါတယ်။ အထူးစွဲတဲ့ Tabel Cell အကွက်လေးများပါဝင်ပါသည် အဲဒီအခါမျိုးမှာ `<td>` သို့ `<th>` အသုံးပြုပါသည်။ အောက်က ပုံကိုအရင်တစ်ချက် ဖြည့်ပေးပါ။

| No | Item | Qty | Price | Total |
|-------------|------|-----|-------|-------|
| 1 | Book | x1 | 1000 | 1000 |
| Total Price | | | | 1000 |

2

Table Row `<tr>`

1

Table တစ်ခုလုံးကို `<table>` Element အဖြစ်သုံးပါတယ်

| No | Item | Qty | Price | Total |
|-------------|------|-----|-------|-------|
| 1 | Book | 1x | 1000 | 1000 |
| Total Price | | | | 1000 |

Table Row (၃) ခုရှိတဲ့
အတွက် `<tr>` ရုံးပါ

3

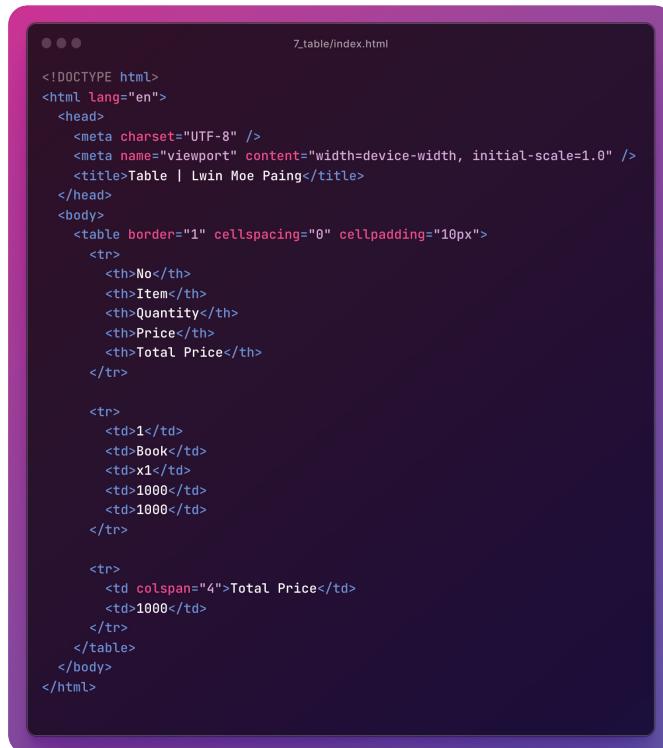
Row အတွင်း မှာ Cell တွေရှိပါတယ် `<td>` သို့ `<th>` ကိုအသုံးပြုပါတယ်

| | | | | |
|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|
| No | Item | Qty | Price | Total |
| <code><td></code> | <code><td></code> | <code><td></code> | <code><td></code> | <code><td></code> |

အရင်ဆုံး ကျွန်တော်က “7_table” ဆိုတဲ့ Folder (ဖိုလ်ဒါ) လေးတည်ဆောက်လိုက်ပါ မယ်။ ပြီးတော့ “index.html” ဆိုတဲ့ HTML ဖိုင်လေး တည်ဆောက်မယ် ပြီးတော့ အောက်က Code အတိုင်း ရေးသားလိုက်ပါမယ်။ အရင် စာမျက်နှာက ပုံအတိုင်းထွက် အောင် ရေးသားခြင်းပါ။

Gist Code ကိုအောက်က လင်နဲ့ ကနေသွားယူနိုင်ပါသည်။

<https://gist.github.com/lwinmoepaing/f5f8feba3afae6fe87fedfffc67dc485>



```

    ...
    7_table/index.html

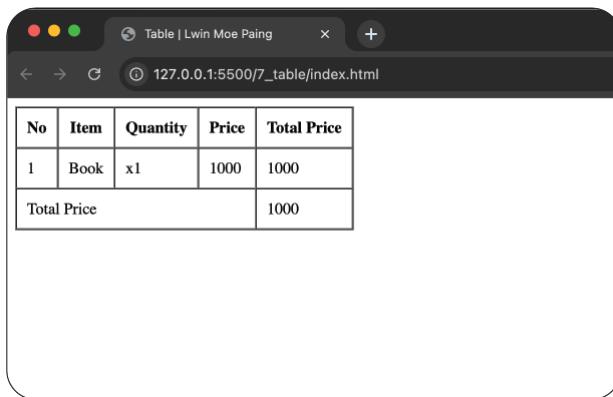
    <!DOCTYPE html>
    <html lang="en">
        <head>
            <meta charset="UTF-8" />
            <meta name="viewport" content="width=device-width, initial-scale=1.0" />
            <title>Table | Lwin Moe Paing</title>
        </head>
        <body>
            <table border="1" cellspacing="0" cellpadding="10px">
                <tr>
                    <th>No</th>
                    <th>Item</th>
                    <th>Quantity</th>
                    <th>Price</th>
                    <th>Total Price</th>
                </tr>

                <tr>
                    <td>1</td>
                    <td>Book</td>
                    <td>x1</td>
                    <td>1000</td>
                    <td>1000</td>
                </tr>

                <tr>
                    <td colspan="4">Total Price</td>
                    <td>1000</td>
                </tr>
            </table>
        </body>
    </html>

```

အောက်က ပုံအတိုင်း Browser ပေါ်တွင် ပေါ်နေပါလိမ့်မည်။



ပထမဆုံး `<table>` Element ကိုမြင်ရပါမယ်။ အထဲတွင်ရှိသော Attribute တွေကိုအရင် လေ့လာကြည့်မယ်နောက် `border="1"` ဆိုတာက table တစ်ခုလုံးအတွက် မျဉ်းကြောင်း Border ကို "1px" လောက်ပေါ်စေချင်လို့ သုံးထားတာပါ။ ဖုန်းတွေဝယ်တဲ့အခါ ကင်မရာ က ဘယ်လောက် MegaPixels (မက်ဂါပစ်လဲ) ပါတယ်ဆိုတာမျိုး ကြားဖူးနောက်ဖြစ်ပါပေါ်မယ်။ ဒီနေရာမှာ px ဆိုတာ Pixel ကိုပြောတာပါ Unit Measurement (ယူနစ်အတိုင်းအတာတစ်ခုပါပဲ)။ သူကတော့ Screen တစ်ခုရဲ့ Image တွေ (သို့) Content တွေကိုပြုပို့အတွက် အသေးဆုံး Unit တစ်ခုပါပဲ။ CSS အပိုင်းကျေရင် units အတိုင်းအတာတွေကို အသေးစိတ် လေ့လာသွားကြရမှာပါ။

နောက်ထပ် Attribute တွေဖြစ်တဲ့ cellpadding ကတော့ border သတ်မှတ်ပြီးတဲ့နောက်မှာ အထဲက Cell အကွက်လေးတွေမှာလဲ Border မျဉ်းကြောင်းများ သတ်မှတ်ပြီးသားဖြစ်သွားပါပြီ အဲမျဉ်းနဲ့ Cell အကွက်အတွင်းမှာရှိတဲ့ Content (သို့) စားသားနဲ့ ကြားထဲအကွာအဝေးကို သတ်မှတ်တာဖြစ်ပါတယ်။ စာဖတ်သူတို့က အဲနေရာမှာ အမျိုးမျိုးထည့်သွင်းပြီး စမ်းသပ်နိုင်ပါတယ်။

ဥပမာ cellpadding="10px" အစား cellpadding="20px" ကဲ့သို့ပြောင်းလည်း စမ်းကြည့်နိုင်ပါတယ်။

cellspacing က လည်း ထိန်ည်းတူပါပဲ အထဲက cell ကွက်တစ်ခုချင်းဆီရဲ့ ကြားအကွာအဝေးပါ။ ဆိုလိုချင်တာက ခုကာက cellpadding က မျဉ်းကြောင်းနဲ့ အကွာအဝေးဖြစ်ပြီး အခုက် cell တစ်ကွက်ချင်းဆီကြားအကွာအဝေးဖြစ်ပါတယ်။

စာဖတ်သူတို့က cellspacing="0" အစား cellpadding="10px" ကဲ့သို့ပြောင်းလည်း စမ်းကြည့်နိုင်ပါတယ်။

မှတ်ချက်။ ။ cellpadding, cellspacing နဲ့ border တို့ကို Attribute အနေနဲ့ အပြင်က Realworld Project တွေမှာ `<table>` element အတွင်း ထည့်သုံးလေ့မရှိပါဘူး။ CSS အခန်းရောက်လျှင် ကျွန်ုတ်တော်တို့က Style တွေကို CSS ဖြင့် Class များခွဲကာအသုံးပြုဖြစ်သွားမှာပါ။

ကျွန်ုတ်တို့ ပထမဆုံး ပြခဲ့တဲ့ ပုံလေးအတိုင်းပါပဲ နောက်ထပ် Row တွေကိုသတ်မှတ်တဲ့အခါ `<tr>` Element ကိုအသုံးပြုပါတယ်။ အတွင်းပိုင်းက Cell အကွက်တွေအတွက် `<td>` ဆိုမဟုတ် `<th>` ကိုသုံးပါတယ်။ `td` ရဲ့ အရှည့်ကောက်က Table Data Cells ပဲဖြစ်ပြီး `th` ရဲ့ အဓိပ္ပာယ်ကတော့ Table Header Cells ပါ။ ဆိုတော့ ကျွန်ုတ်တို့ ပထမ `<tr>` row အတွင်းပိုင်းမှာ header cell အနေနဲ့ဖြစ်တာကြောင့် `<th>` တွေကိုအသုံးပြုထားတာကို သတိပြုမိမှာပါ။ ဒါကြောင့် နောက်ထပ် row တွေအတွက်တော့ `<tr>` `<td>` ကိုသုံးသွားခဲ့ပါတယ်။

အောက်ဆုံး row မှာ တော့ လုံးဝမတူတဲ့ Cell အကွက် (၄) ကွက်စာနေရာယူသွားတာ ကတော့ Total Price ပဲဖြစ်ပါတယ်။ ဘူး <td colspan="4"> ဆိုတာမှာ Attribute ဖြစ်တဲ့ colspan ရဲ့ ဆိုလိုရင်းက Column (ကော်လုံး) နေရာ (၄) ခုစာယူပါမယ်ဆိုပြီး သတ်မှတ်လိုက်တာဖြစ်ပါတယ်။ အောက်က ပုံလေးကို တစ်ချက် ကြည့်ပေးပါ...

| | | | | |
|------------------|------|----|------|------|
| 1 | Book | 1x | 1000 | 1000 |
| <td colspan="4"> | | | | |

ဒီပုံလေးကို ကြည့်ပြီး တဲ့အခါမှာတော့ Column နေရာယူပုံကို ကြည့်ပြီး နားလည်လိမ့်မယ်လို့ ယူဆပါတယ်ခင်ပျော်။ ဒါဆို ရင် Row ရဲ့ နေရာယူတဲ့ဟာကိုလည်း နားလည်သွားလိမ့်မယ်လို့ မျော်လင့်မိပါတယ်။ Col အတွက် colspan နဲ့ နေရာယူနိုင်သလို Row အတွက်လည်း rowspan နဲ့ နေရာယူနိုင်ပါတယ် စာဖတ်သူတို့က အောက်က Table လေးကို ကိုယ်တိုင်လုပ်နိုင်လိမ့်မယ်လို့ မျော်လင့်မိပါတယ်။

အရင် ကိုယ်ရဲ့ Local PC မှာဖြစ်ဖြစ် အရင်စမ်းပြီး ရေးကြည့်ပါ။ ကျွန်ုတ်ကတော့ "7_table" ရဲ့ Folder (ဖိုလ်ခါ) အတွင်းမှာ နောက်ထပ် "row.html" ဆိုပြီး HTML ဖိုင်တည်ဆောက်ပြီး ရေးစေချင်ပါတယ်။

| | | | |
|----------|---------|-------|----------|
| No | 11 AM | 12 PM | 1 PM |
| 1 | Reading | Lunch | Riding |
| 2 | Playing | | Swimming |
| Sleeping | | | |

စာဖတ်သူတို့ ရေးပြီး တဲ့အခါ နောက်စာမျက်နှာမှာ အဖော်တိုက်ကြတာပေါ့။

Gist Code ကိုအောက်က လင်း ကနေသွားယူနိုင်ပါသည်။

<https://gist.github.com/lwinmoepaing/b0ee85f495da8dffcac774b4c7b8d3ee>

```
7_table/row.html
...
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Table RowCol | Lwin Moe Paing</title>
</head>
<body>
    <table border="1" cellspacing="0" cellpadding="10px">
        <tr>
            <th>No</th>
            <th>11AM</th>
            <th>12PM</th>
            <th>1PM</th>
        </tr>

        <tr>
            <td>1</td>
            <td>Reading</td>
            <td rowspan="2">Lunch</td>
            <td>Riding</td>
        </tr>

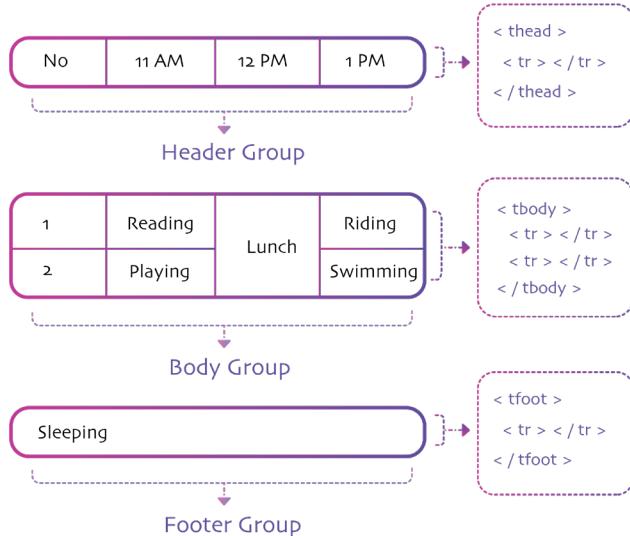
        <tr>
            <td>2</td>
            <td>Playing</td>
            <td>Swimming</td>
        </tr>

        <tr>
            <td colspan="4">Sleeping</td>
        </tr>
    </table>
</body>
</html>
```

စာဖတ်သူတို့လည်း တွေးပြီး ရေးနိုင်ခဲ့မှာ သေချာပါတယ်။ ကျွန်တော်တို့ စဉ်းစားစရာက ဒုတိယ Row `<tr>` ထဲမှာရှိတဲ့ Cell တစ်ကိုကာသာ Row ကို (၂) နေရာယူယူပါ အဲ တော့ Lunch ရှိတဲ့ `<td rowspan="2">Lunch</td>` ဆိုပြီး ထို Cell ကိုပဲ (၂) နေရာစာကို ရေးပေးလိုက်ရင် ရပါပြီ။ အောက်ဆုံး row ကတော့ ရှိသမျှ Column အကွက်ကို ယူထားတဲ့အတွက် `colspan` ကို (၄) နေရာစာ ယူပေးလိုက်ရမှာပါ။

Table နဲ့ပါတ်သတ်ပြီး ဒီလောက်ဆိုတော်တော်လေး အခြေခံအားဖြင့် သိသွားကြပါပြီ။ သို့ပေမယ့် အယားကွက်တွေအတွက် Group လိုက်ရေးထုံးမျိုးလည်း နောက်ပိုင်း အသုံးပြုလာကြပါတယ်။ `<thead>`, `<tbody>` နဲ့ `<tfoot>` ဆိုတဲ့ Element (၃) ခုပါ ခုကာက ရေးခဲ့တဲ့ Code ကိုပဲ ပြန်လည်အသုံးပြုပြီး ဘယ်လိုရေးသားရမလဲ လေ့လာ ကြည့်ရအောင်။

Header Group အတွက်ကို <thead> ကိုသုံးမှာဖော်ပြီး။ Body Group အတွက်ကို <tbody>၊ Footer Group အတွက်ကို <tfoot> ကို အသုံးပြုမှာပါ။



အောက်က Code ကို ကြည့်လိုက်ရင် ပိုရင်းသွားမယ်ထင်ပါတယ်။ Codepen Link ကို အောက်မှာ ပေးထားပါသည်။

<https://codepen.io/lwinmoeapaing/pen/rNQmVNL>

```
7_table/group_table.html
● ● ●
<table border="1" cellspacing="0" cellpadding="10px">
  <thead>
    <tr>
      <th>No</th>
      <th>11AM</th>
      <th>12PM</th>
      <th>1PM</th>
    </tr>
  </thead>

  <tbody>
    <tr>
      <td>1</td>
      <td>Reading</td>
      <td rowspan="2">Lunch</td>
      <td>Riding</td>
    </tr>

    <tr>
      <td>2</td>
      <td>Playing</td>
      <td>Swimming</td>
    </tr>
  </tbody>

  <tfoot>
    <tr>
      <td colspan="4">Sleeping</td>
    </tr>
  </tfoot>
</table>
```

Form Elements များ

Form တွေက အရေးကြီးပါတယ် Web page တော်တော်များများမှာ Login လုပ်ရတာ Register လုပ်ရတာတွေကို စာဖတ်သူတွေက တွေ့ဖူးနေကြဖော်မှာပါ။ အဲလိုအရာတွေက <form> ဖော်တွေဖြည့်ရတဲ့ သဘောတရားပါပဲ။

Element <form> ကို အမိက အနေနဲ့ wrap (အုပ်) ပြီးရေးသားပါတယ်။ ကျွန်တော်တို့ အပြင်လောက မှာ ဖော်ဖြည့်ဖူးတာမျိုး၊ ရိုကြဖူးမှာပါ ဥပမာ - ကျောင်းဝင်ခွင့်ဖောင်ပဲဖြစ်ဖော် ရည်ရွယ်ချက်ရှိဖြည့်ကြရတာမျိုးပါ

အဲလိုပဲ HTML က <form> Element မှာလဲ action ဆိတဲ့ရည်ရွယ်ချက် Attribute ကို လိုအပ်ပါတယ် သူက ဘယ်ဖောင်ကို ဖြည့်မှာလဲဆိတဲ့ တည်နေရာကို ထည့်ပေးရတာပါ။ သူက URL ဖြစ်နိုင်သလို Server Side Script လည်းဖြစ်နိုင်ပါတယ်။ action (attribute) ကို ထည့်မပေးလိုက်ရင်တော့ Default အနေနဲ့ လက်ရှိရောက်နေတဲ့ Page ကိုပဲ Action ခေါ်သွားမှာပါ။

Example <form action="/login"> လို့ထည့်ပေးလိုက်ရင် Login ဆိတဲ့ Route ကိုသွားပြီး (action) လုပ်ဆောင်မယ့်ရည်ရွယ်ချက်ပါ။

နောက်ထပ် Attribute အနေနဲ့ method ဆိတဲ့ရှိပါတယ် သူရဲ့ Value အနေနဲ့ "get" method နဲ့ "post" method ကိုထည့်သွင်းရေးလေ့ရှိပါတယ်။ "get" method ကို တော့ Server ဘက်က Data တွေကို ရယူပဲအခါ အသုံးများပြီး၊ "post" method ကတော့ Server ဘက်ကို ဒေတာတွေကိုပေးပိုပြီးတော့ တခုစုလုပ်ဆောင်စေချင်တဲ့အခါ မှာအသုံးများပါတယ်။

ဥပမာ အပြည့်အစုံ <form action="/login" method="post"> ဆိတဲ့ ပုံစံနဲ့ ရေးသားလိုက် ခြင်းသည် "/login" ဆိတဲ့ url ကိုသွားပြီး Action လုပ်ပါမယ် Method ကတော့ Post ဖြစ်တာကြောင့် Server ဘက်ကနေ Login ဆိတဲ့လုပ်ဆောင်ချက်ကို လုပ်ဆောင်ပေးပါဆိုတဲ့ သဘောမျိုးဖြစ်ပါတယ်။

အပြင်လက်တွေ့မှာ ဖောင်တစ်ခုကိုဖြည့်ရတော့မယ်ဆိုရင် ကျွန်တော်တို့ စာရွက် စာတန်းတွေထဲမှာ ဖြည့်စရာ ဂွက်လပ်တွေ၊ ရွေးချယ်စရာ ဖြင်တွေရှိလေ့ရှိပါတယ်။ ဥပမာ နာမည် ဆိုပြီးဂွက်လပ် ထားထားရင် ကိုယ့်နာမည် ရေးပေးရတာမျိုးဖြစ်ပြီး အိမ်ထောင် - လူပျိုး (သို့) အိမ်ထောင် ဆိတဲ့လေးထောင့်ဂွက်ကို အမှန်ခြင်ပေးရတာမျိုး တွေပါပါတယ်။ အဲလိုမျိုးပါပဲ Web form မှာလည်း အဲလိုမျိုးကို input form element လို ခေါ်ပါတယ်။

Form Control များ

<form> Element အတွင်းမှာ Form Controls Type များစွာပါဝင်ကြပါတယ်။ အများက Input Form Element တွေဖြစ်တဲ့ <input> Element ကိုအသုံးများပါတယ်။ အခြား <select>, <label>, <textarea> အစရှိတဲ့ Element များတို့လည်း သုံးပါတယ်။ ကျွန်တော်တို့ Website ကိုလာရောက် လည်ပတ်တဲ့ အသုံးပြုသွေဆီကနေ Data တွေ ကို Collect ကောက်ယူစွဲအတွက် Form Controls တွေကို အသုံးပြုကြရပါတယ်။ Common အသုံးများတဲ့ Form Control အများစုံကို အောက်က ပုံနှင့်လောက်ည့်ရအောင် ...

အသုံးများသော Form Control များ

စာသားထည့်ခြင်းများ

Input Text

တစ်ကြောင်း စာသားထည့်ခြင်း

<input type="text" />

Lwin Moe Paing

Input Password

စကားဂုဏ်စာသားထည့်ခြင်း

<input type="password" />

Text Area [multiline]

တစ်ကြောင်းထောက်ပို့စာသားထည့်ခြင်း

<textarea> </textarea>

Enter your comments...

ရွေးချယ်ခြင်းများ

Input Radio

တစ်ခုတည်းရွေးချယ်ခြင်း

<input type="radio" />

ကျော် မှု အမြဲး

Input Checkbox

တစ်ခုထက်ပို့ရွေးချယ်ခြင်း

<input type="checkbox" />

FB Instra Twitter

Select Box

selector ဖြင့် Dropdown ရွေးချယ်ခြင်း

<select>

Yangon



Form Submit တင်ခြင်း

Submit Button

Form အတည်ပြုခြင်း

<input type="submit" />

<button type="submit" />

SUBMIT

File Upload တင်ခြင်း

File Input

စိုင်ကို Upload တင်ခြင်း

<input type="file" />

File Name

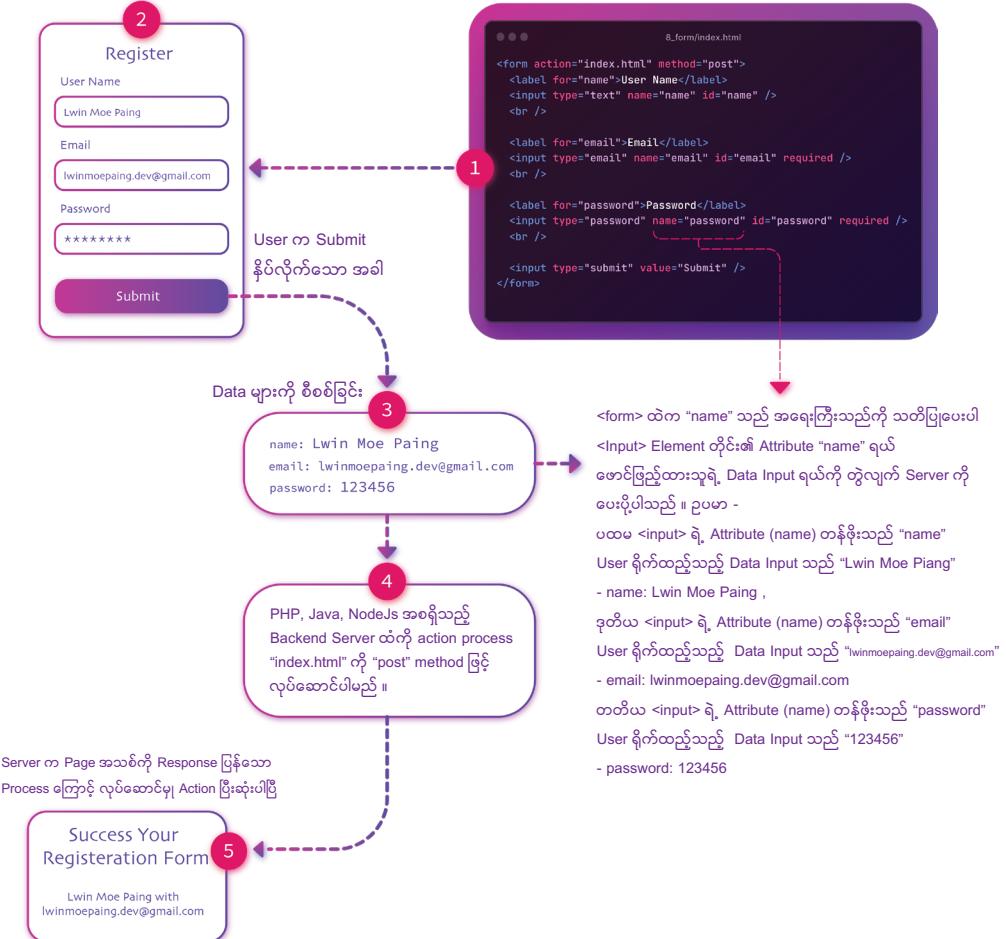
Browse...

Upload

Form တွေ ဘယ်လိုအလုပ်လုပ်လဲ ?

Browser ပေါ်ပြင်တွေ ရပါမြို့

ရေးသားထားသော Form HTML Element



အညွှန်း (၂) အရ User က Submit လုပ်လိုက်တဲ့အခါ ရေးသားထားသော Form Control Element များ ဖြစ်သော <input>, <select>, <textarea> Elements များ၏ Attribute (name) ရယ် User က ရိုက်ထည့်လိုက်သော တန်ဖိုးများကို အညွှန်း (၃) အရရှစ်စိပြုး Server ထံသို့ Action Process ဆောင်ရွက်စေပါသည်။

ပြီးသည့်နောက်မှ Server ၏ Response ပြန်ချက်အရ Page အသစ်ကို ကျွန်ုတ်တို့ ပြန်လည်ပြသပေးပါသည်။

ပထမဆုံး အခန်းမှာ Traditional Website တွေဟာ Request , Response ပြင့် လုပ်ဆင် ကြေသည်ကို ဖော်ပြထားခဲ့ပြီးလည်းဖြစ်ပါသည်။

Text Input တွေကိုလေ့လာကြည့်မယ်

အသုံးပြုတွေတွေက Keyboard ကနေရှိက်ထည့်လို့ရမယ် Text Input

(Form Control) တွေမှာ <input> Element ကိုသုံးပါတယ် သူ့ရဲ့ Attribute တစ်ခုဖြစ်တယ် type ရဲ့ တန်ဖိုးတွေကိုပြောင်းလဲပြီး လိုအပ်တဲ့ အလိုရှိတဲ့ Form ကိုရေးသားသတ်မှတ်ပေးနိုင်ပါတယ်။

<label> ဆိုတဲ့ element ကို Input ကို Focus လုပ်ပေးစီရန်နဲ့ Form Control ကို Label ခေါင်းစဉ်ပေးတဲ့နေ့ရာမှာ သုံးပါတယ်။ အောက်က ပုံကိုလေ့လာကြည့်ရအောင်

<label for="element_id"> User Name </ label>

Label ဘာဘာကိုယ့်အသုံးပြုပါတယ်။ Label ကိုနိုင်တဲ့အသုံးပြုမှုများ

ပါတယ်။

<input type="text" name="form_name" id="element_id" required />

text
password
email
number
tel
date

Server ကို Action Process

ပေးပို့ခြင်းအတွက်
Form ရဲ့ Attribute (name)
ကိုသတ်မှတ်ရန်လိုပါတယ်

HTML Attribute တွေရာ value
ပါဝါယူလုပ်မယ် ဒါ Required မှု
တော့ Attribute name တစ်လုံး
တည်းအသုံးပြုထားတာလို့ သတိ
ဖြပ်ပေးပါ။

Form Submit လုပ်တဲ့အသုံးပြုမှု
required ပါခဲ့ရင် အသုံးပြုသူက
ဖောင်ကို ဖြည့်ပေး၊ ထည့်ပေးနို့
လိုကိုလိုပြုပြောင်း သတ်မှတ်တဲ့အခါ
အသုံးပြုပါတယ်။

အညွှန်း (၄) အရ type Attribute မှာ text, password, email, number, tel, date ဆိုပြီး
စာဖတ်သူတို့က လိုအပ်တဲ့ Type ကိုပြောင်းလော်အသုံးပြုနိုင်ပါတယ်။ text ဆိုရင်စာသား၊
password ဆိုရင် စာသားတွေကို * များဖြင့်ကွယ်ပေးထားမှာဖြစ်ပါတယ်။ email က
Email များအသုံးပြုရင်သုံးတယ်၊ number က နံပါတ်တွေပုံက်ထည့်စေချင်တဲ့အခါမျိုး
မှာအသုံးပြုပါတယ်၊ tel က တော့ တယ်လိုနှင့်နံပါတ်တွေ ထည့်စေချင်ရင်သုံးပါတယ်၊
date ကတော့ ရက်စွဲတွေကို ထည့်စေချင်တဲ့အခါအသုံးပြုပါတယ်။

နောက်ထပ် Multiline Input တွေ ရိုက်ထည့်စေချင်ရင်တော့ <textarea> ဆိုတဲ့ Element
ကိုအသုံးပြုသွားမှာပါ။ နောက်တစ်မျက်နှာမှာ textarea အသုံးပြုပုံကိုလေ့လာကြပါမယ်။

<input /> Element အမျိုးအစားကတော့ Self Closing Element ဖြစ်ပါတယ် ">" ဒါနဲ့ အဆုံးသတ်ပါတယ်။ <textarea> Element မှာက Self Closing Element မဟုတ်ပဲ သူ မှာက အပိတ် Tag (Closing Tag) ပါတာကို သတိပြုပေးပါ။ ရေးသားပုံကို အောက်က Code ကို လေ့လာကြည့်နိုင်ပါသည်။

```
<label for="note_element_id"> Note </label>

<textarea name="form_name" id="note_element_id" ></textarea>
```

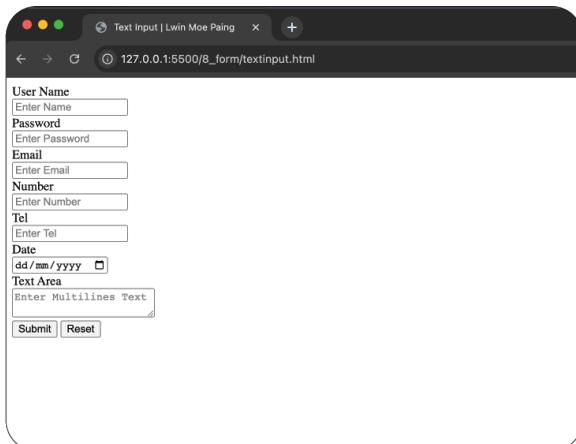
Text Input Element တွေ ကို အခြေခံအားဖြင့် သိသွားကြပြီလို့ယူဆပါတယ်။ ဆိုတော့ ကျွန်တော်တို့ လက်တွေ၊ Coding ပိုင်းကို လေ့လာကြည့်ဖို့အတွက် အရင်ဆုံး Folder (ဖို့လိုအပ်) အသစ်တည်ဆောက်ရအောင်။ နာမည်ကို "8_form" လို့ ကျွန်တော်က ပေးလိုက်ပါမယ်။ ပြီးတော့ "textinput.html" ဆိုပြီး HTML file (ဖိုင်) အသစ်တစ်ခု တည်ဆောက်လိုက်ပါမယ်။ ပြီးနောက် အောက်က Gist Code လေးကို စာဖတ်သူတို့က ကူးယူ ပြီး Browser မှာဖွင့်ကြည့်လိုက်ပါ။

<https://gist.github.com/lwinmoepaing/09ee3c7cd9b70f45ac96eb09d56b1bf8>

Code Pen အတွက်အောက်က လင့်ခိုကနေစမ်းလိုရပါတယ်

<https://codepen.io/lwinmoepaing/pen/zYMwzz>

Browser ဘုင်အောက်က ပုံအတိုင်း ပေါ်လာပါမည်။



ကုတ်တွေကို ဝင်စမ်းပြီး Run ကြည့်ပြီးရင် စာဖတ်သူတို့ Browser မှာလည်း အဆင်ပြေ
လိမ့်မယ်လို့ မျှော်လင့်ပါတယ်။ အခုက အသစ်အနေပါလာတာက Attribute name
placeholder ပဲဖြစ်ပါတယ်။ သူက Input Form မရှိက်ခင်မှာ ဖော်ပြုမယ့် စာသားအဖြစ်
ကြိုတင်သတ်မှတ်ပေးတဲ့ နေရာမှာ အသုံးများပါတယ်။

နောက်ထက် Text Input ပဲမဟုတ်တဲ့ type (J) မျိုးက တစ်ခုက submit ဖြစ်ပြီးသူက
Form ကို Submit လုပ်တော့မယ့် button လုပ်ဖန်တီးတဲ့ နေရာမှာသုံးပြီး။
reset ကတော့ Form တစ်ခုလုံးကို Clear သွားအောင် reset ပြန်ချုတဲ့ နေရာမှာ
အသုံးပြုပါတယ်။

Selector ရွှေချယ်ရာတွင်သုံးမယ့် Form Controls များ

အသုံးပြုတွေက ဖောင်ဖြည့်ရမှာ Keyboard (Input) တရာ့ခုကနေ ရိုက်ထည့်စရာ
မလိုဘဲ ရွှေးချယ်လိုရတဲ့ CheckBox , Radio Button တွေအတွက် <input> Element
ကိုပဲအသုံးပြုပါတယ်။ Dropdown နဲ့ အတွက်ကျေတွေ့ <select> ဆိုတဲ့ Element ကို
သုံးပါတယ်။ အရင်ဆုံး Radio Button ကို တစ်ခုပဲရွှေးချယ်တဲ့ အခါမှာသုံးပါတယ်။
ဥပမာ - Gender ရွှေးချယ်ပါပေါ့ ကျေး (သို့) မ (သို့) အခြား
ရည်ရွယ်ချက်သည် ကိုယ်က Gender တစ်မျိုးပဲဖြစ်နိုင်လိုပါ။ အဲလိုအခါမျိုးမှာ
Radio ရွှေးချယ်မှုကိုသုံးလေ့ရှိပါတယ်။ နောက်ထပ်ဥပမာအရပေးရမယ်ဆိုရင် ကိုယ့်ရဲ့
WebSite ကို Theme ရွှေးခိုင်တာမျိုးပေါ့ Light Theme (အလင်း) သို့မဟုတ်
Dark Theme (အမျှင်) ရွှေးခိုင်းတာမျိုးတွေ့မှာ သုံးလေ့ရှိပါတယ် On , Off ဖွင့်ရင်ဖွင့်
သို့မဟုတ် ပိတ် Setting တွေမှာလည်းတွေ့လေ့ရှိပါတယ်။ အောက် Radio Input အတွက် Code ကိုလေ့လာကြည့်ရအောင် ...

```
<input type="radio" name="gender" value="male" id="male" />
<label for="male"> Male </ label>

<input type="radio" name="gender" value="female" id="female" />
<label for="female"> Female</ label>
```

သတိထားရမှာက Attribute ဖြစ်တယ့် name ရဲ့ တန်ဖိုး gender ကို အတူတူပဲထား
ထားတာကို သတိပြုရမှာပါ။ အစကလေ့လာခဲ့သလို Server အတွက် Form ရဲ့
Attribute (name) ကအမြဲပါစို့လိုပါတယ်။ Radio Element မှာကျေတွေ့ name က
Group လိုသဘောမျိုးပါ Attribute name Gender အတွက် Value ကတော့ အသုံးပြု
သူ ရွှေးချယ်လိုက်တာကိုပဲ ပို့ဂိုက်မှာပါ။ အဲအတွက်မို့ <input> Element ထဲမှာ value
တန်ဖိုးတွေ ကြိုးသတ်မှတ်ပေးထားစို့လိုပါတယ်။

အလားတူပါပဲ နောက်ထပ် Checkbox ပုံစံနဲ့ ရွေးချယ် စရာ <input> Element တစ်ခုရှိပါတယ်။ သုကတေဘာ့ တစ်ခုထပ်ပို ရွေးချယ်တဲ့အခါမှာ သုံးလေ့ရှိပါတယ်။

ဥပမာ သင်ဘာသာဝကား ဘယ်နှစ်မျိုးရလဲ ? ဆိုရင် ပေးထားချက်ရှိတယ်ဆိုပါဆို English, Myanmar, Chinese ဆိုရင် ၂ မျိုးရဲ့ရင် ၂ မျိုးလုံးရွေးချယ်ပေးတာမျိုးကို ဆိုလိုတာပါ။ အောက်က Radio Input အတွက် Code ကိုလေ့လာကြည့်ရအောင်။

```
<input type="checkbox" name="language" value="en" id="en" />
<label for="en"> English </ label>

<input type="checkbox" name="language" value="cn" id="cn" />
<label for="cn"> Chinese </ label>

<input type="checkbox" name="language" value="my" id="my" />
<label for="my"> Myanmar</ label>
```

ဒီမှာလဲ Attribute ဖြစ်တယ် name ရဲ့ တန်ဖိုး language ကို အတူတူပဲထားထားတာကို သတိပြုရမှာပါ။ Selector Form Controls တွေမှာ name က Group လိုသဘောမျိုးဖြစ်ပြီး အတူတူဖြစ်နေဖို့လိုပါတယ်၊ ပြီးတော့ value တန်ဖိုးတွေ ကြိုသတ်မှတ်ပေးထားဖို့ လိုပါတယ်။

Dropdown Selector ကို ကျွန်ုတ်တို့ ဆက်သွားရအောင် Checkbox နဲ့ Radio က <input> Element ကို အသုံးပြုပေမယ့်၊ Dropdown အတွက်က <select> Element ကို အသုံးပြုရပါတယ်၊ အောက်က Code ကိုလေ့လာကြည့်ရအောင် ...

```
<select name="region" >
    <option value="YGN"> Yangon </ option>
    <option value="MDY"> Mandalay </ option>
    <option value="SHN"> SHAN </ option>
</ select >
```

သူရေးထုံးက ရှင်းပါတယ် <select> element မှာ Form Controls တွေရဲ့ သဘောတရားအတိုင်း Attribute (name) ပါပါတယ်။ အထဲက Child Elements က <option> element နဲ့အတူ သူအထဲမှာ value (Attribute) ပါပါတယ် ပြီးတော့ အသုံးပြုသူတွေဖြင့်ရမယ့် စာသားက option element ရဲ့content ပါ။ နောက်တစ်မျက်နှာမှာ Code တွေ သွားစမ်းပါတော့မယ်။

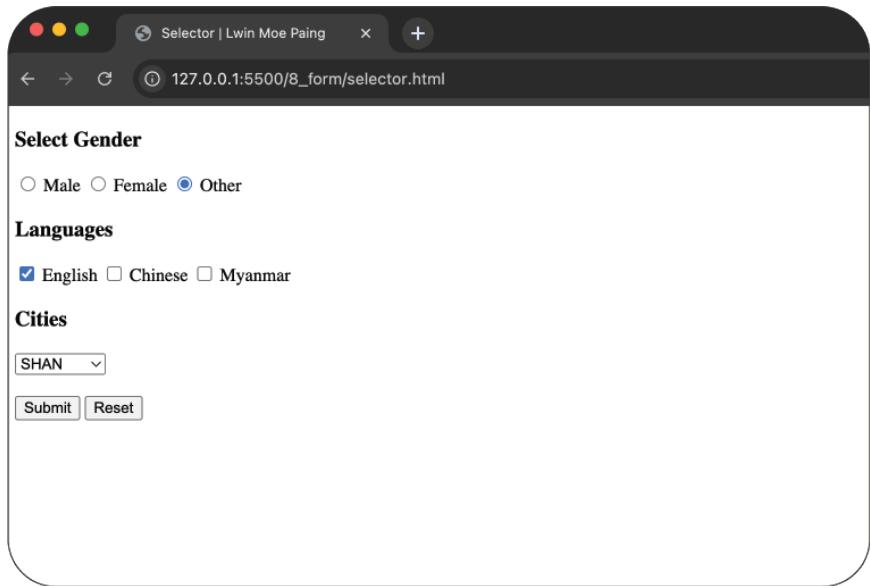
Selector (ရ) မျိုးကို လေ့လာပြီးသွားပါပဲ။ Radio , Checkbox နဲ့ Dropdown ပါ။ အခု လက်တွေ၊ Coding ပိုင်းကို လေ့လာကြည့်ဖို့အတွက် အရင်ဆုံး Folder (ဖို့လုပ်ပါ) နာမည် “8_form” ထဲမှာပဲ “selector.html” ဆိုပြီး HTML file (ဖိုင်) အသစ် တစ်ခု တည်ဆောက်လိုက်ပါမယ်။ အောက်က Gist Code လေးကို စာဖတ်သူတို့က ကူးယူပြီး Browser မှာဖွင့်ကြည့်လိုက်ပါ။

<https://gist.github.com/lwinmoeapaing/2f1c2c972e3f5aa08a1cf235ce70b99e>

Code Pen အတွက်အောက်က လင့်ခိုက်နေစမ်းလို့ရပါတယ်

<https://codepen.io/lwinmoeapaing/pen/jOQmGxW>

Browser ဘွင်အောက်က ပုံအတိုင်း ပေါ်လာပါမည်။



ဒီအပိုင်းမှာ တွေ့လာရတာက <input> Element ထဲမှာ Attribute name “checked” ဆုံးတာက Default Value ကိုကြိုပေးထားတာပါ အသုံးပြုသူတွေ မရွေးရသေးခင် ကိုယ်က ကြိုပြီး ထည့်ပေးထားတာမျိုးပါ အပေါ်ကပုံမှာဆိုရင် English နဲ့ Other က ကြိုပြီး သတ်မှတ်ထားတာဖြစ်ပါတယ်။

<select> Element ထဲမှာလည်း ထိနည်းတူစွာပါပဲ သူကျတော့ အတွင်းက <option> element ထဲမှာပဲ Attribute သတ်မှတ်ပေးရတာပါ သူကျတော့ “selected” ပါ သတိထား စေချင်ပါတယ်။

Form Submit Element

Form Controls တွေတိုင်းမှာ အသုံးများခဲ့တဲ့ `<input>` Element ပါပဲ။ type Attribute ရဲ့
တန်ဖိုးကာသာ "submit" လိုပေးပြီး value (Attribute) ကိုတော့ ပေါ်စေချင်တဲ့ စာသားကို
ရေးသားပေးလိုက်ရှုပါပဲ။

ဥပမာ - `<input type="submit" value= "submit စာသား" />`

နောက်တစ်နည်းကတော့ `<button>` Element ကိုအသုံးပြုတာပါ အောက်က Code ကို
တစ်ချက်လောက် ကြည့်ကြည့်ပါ။

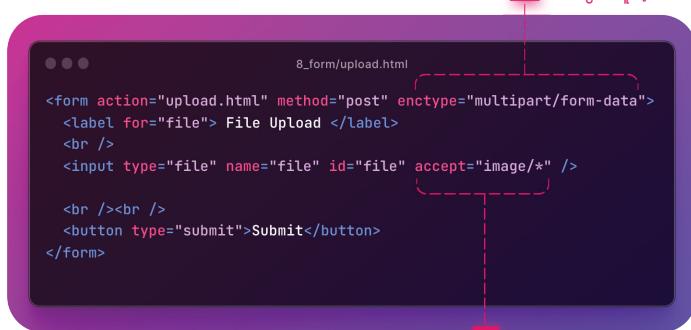
```
<button type="submit"> Submit </button>
```

Attribute (type) မှာ "submit" ကိုသုံးပြီး Content ကို ဖော်ပြချင်တဲ့စာသားရေးပြီး
အသုံးပြလို့ရပါတယ်။ `<button>` Element အတွက် Attribute (type) မှာ "button"
ဖြင့်သာ အသုံးပိုများပါတယ်။

Form Upload Element

Form Controls တွေထဲမှာ File တွေကို Upload လုပ်တာလည်းပါလာမှာပါ။ အဲဒီ
အခါမှာလည်း `<input>` Element ကိုသာအသုံးပြပြီး type Attribute ကိုသာ "file" လို့
ပြောင်းလဲအသုံးပြပေးလိုက်ရင်ရပါပြီ။
အောက်က Code ကိုတစ်ချက်လောက် ကြည့်ကြည့်ပါ။

File ဆုံး Upload လုပ်မယ်ဆို Form ကို Encoding System ပြောင်းပေးရပါတယ်။ Multipart/
form-data ဘူး Server ဘက်က File Handle ရဲ့
လွှားကျတဲ့ Encoding တစ်ဖျိုးပါ။



```
8_form/upload.html
<form action="upload.html" method="post" enctype="multipart/form-data">
  <label for="file"> File Upload </label>
  <br />
  <input type="file" name="file" id="file" accept="image/*" />
  <br /><br />
  <button type="submit">Submit</button>
</form>
```

ဒါကတော့ Browse ဖွင့်လိုက်တဲ့အချိန်မှာ အသုံးပြုဘူး
တွေ့ဆုံးကနေ လက်ခံမယ့် File Type အချို့အစားကို
သတ်မှတ်ထားတာပါ။ အခုန့် ဖုံးတွေပဲ လက်ခံမယ်လို့
ဆိုလိုတာပါ

File Upload ကိုလက်တွေ့ Coding ရေးကြည့်ဖို့အတွက် အရင်ဆုံး Folder (ဖို့လိုအပ်) နာမည် “8_form” ထဲမှာပဲ “upload.html” ဆိုပြီး HTML file (ဖိုင်) အသစ်တစ်ခု တည်ဆောက်လိုက်ပါမယ်။ အောက်က Gist Code လေးကို စာဖတ်သူတို့က ကူးယူပြီး Browser မှာဖွင့်ကြည့်လိုက်ပါ။

<https://gist.github.com/lwinmoepaing/8fb034be925031b64bb07586d95eaab9>

Codepen လင့်ခိုလည်း အောက်ကနေဝင်ကြည့်လိုရပါတယ်

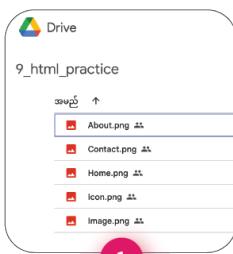
<https://codepen.io/lwinmoepaing/pen/NWEjXEe>

From Theory to Practice - 2 : New HTML Practice

Practice လုပ်ဖို့အချိန်ကိုရောက်ရှိလို့လာပြီဖြစ်ပါတယ်။

Form Controls တွေရယ် Table ထဲမှာပဲထည်ဆောက်ခြင်းတွေကို လေ့လာခဲ့ပြီး ဖြစ်ပါတယ်။ အခုံ Folder ကို “9_html_practice” လိုပဲ ဖို့လိုအပ်နာမည်လေးပေးလိုက်မယ် ယူ။ အဲထဲမှာ “index.html”, “about.html”, “contact.html” ဆိုပြီး file (HTML ဖိုင်) - ၃ ဖိုင် တည်ဆောက်လိုက်ပါမယ် ပြီးရင်၊ လိုအပ်သော Image File တွေကို အောက်က Google Drive Link က နေ Assets တွေကို Download သွားဆွဲပေးပါ။

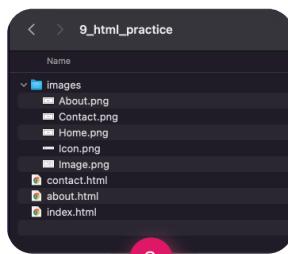
https://drive.google.com/drive/folders/14fvpcosk4KWU9uMFpX3ll9bEUtArFY0e?usp=drive_link



1

Google Drive ထဲရှိ

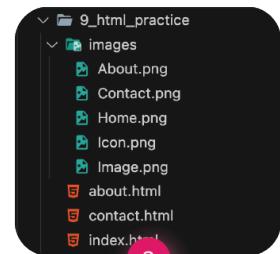
Png ဖိုင် (၅) ဖိုင်ငံ့ေး Download ခွဲပေးပါ



2

ပြုလုပ်ထားသော 9_html_practice

Folder (ဖို့လိုအပ်) ထဲတွင် “images” ဆိုတော်

Folder အသစ်တစ်ခု တည်ဆောက်ပြီး ထို (၅) ဖုံး
လုံး Copy ကူးယူလိုက်ပါမယ်

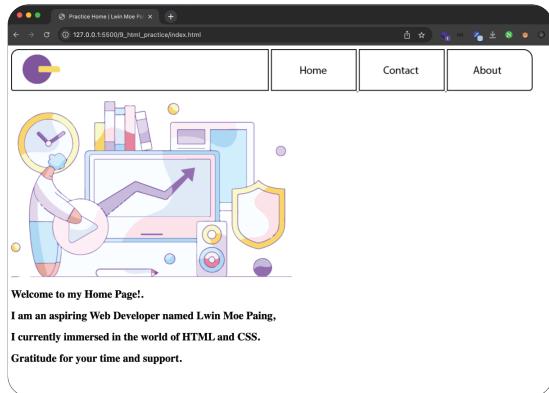
3

VS code အတွင်း

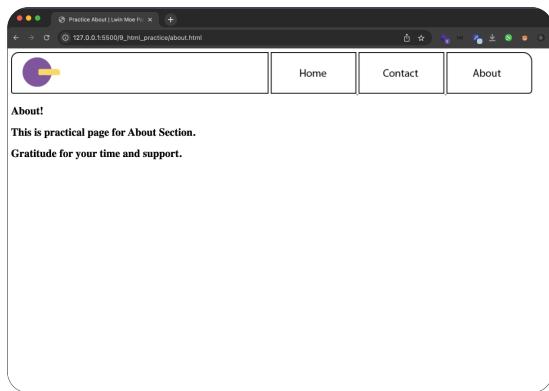
ထိုကဲသို့ ဖြစ်ရပါလိမ့်

နောက်တစ်မျက်နှာမှာ Website Page စာမျက်နှာ (၃) ပုံကိုပေးထားပါမယ်။ ထို ပုံများ အတိုင်းထွက်အောင် စာဖတ်သူတို့က ရေးပေးရမှာဖြစ်ပါတယ်။

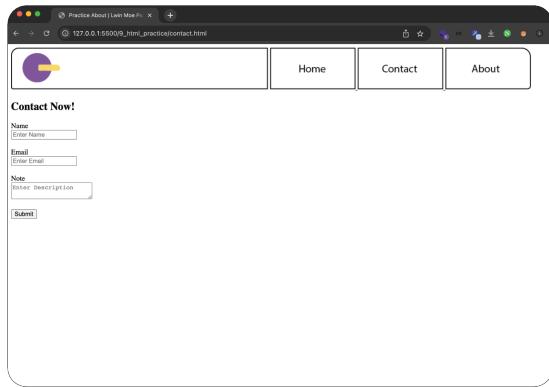
အောက်မှာပေးထားတဲ့ Home Page , About Page နဲ့ Contact Page တို့ကို ချိတ်ဆက်ပေးရမှာဖြစ်ပါတယ်။ စာဖတ်သူတို့ ကိုယ်တိုင် ရေးသားနိုင်လိမ့်ယ်လို့လည်း ယုံကြည်ပါတယ်။ နောက်တစ်မျက်နှာမှာ Code တွေနဲ့ အဖော်ကို ပေးထားပါမယ်။



index.html for Home Page



about.html for About Page



contact.html for Contact Page

Code ကို အောက်က Gist မှာသွားရောက် တိုက်ကြည့်ပြီးလေ့လာနိုင်ပါတယ်။
ကိုယ်ပိုင်ပုံစံဖြင့် Result အရ ကောင်းမွန်စွာတွက်နေ့ပြီး ဖြစ်သော်လည်း ကျွန်တော်ပေး
ထားသော Code နဲ့တူညီမှ မှန်တယ်လို့မဆိုလိုပါ။

index.html အတွက် Gist Code -

<https://gist.github.com/lwinmoepaing/05001bb266f0ddc8f77b116d122e4626>

about.html အတွက် Gist Code -

<https://gist.github.com/lwinmoepaing/c76d3e95fdc751937b32afe9174699ce>

contact.html အတွက် Gist Code -

<https://gist.github.com/lwinmoepaing/f0d81e51ae6864e213f9be8ec6fbb7fb>

အောက်က CodeSandBox မှာလည်း Online ပေါ်တိုက်ရှိက် Run ကြည့်နိုင်ပါသည်။

<https://codesandbox.io/s/9-html-practice-l485nz>

Formatting Elements များလေ့လာခြင်း

<i> - စတဲ့ Elements တွေကတော့ စကာလုံးကို စောင်းစေချင်တဲ့အခါ သုံးပြီး
Italic အရှည်ကောက်ကို ယူထားတာပါ။ ကတော့ Emphasized ရဲ့ အရှည်နှစ်
လုံး em ကိုယူထားတာပါ။

 Element ကို စာလုံး Bold စကားလုံးထူထူလေးဖြစ်စေချင်တဲ့အခါ အသုံးပြုပါ
တယ်။

 Element လည်း အတူတူပါဝါ Important Text ဖြစ်တဲ့အခါ Bold ပုံစံအတိုင်း
ဖော်ပြလို့ရပါတယ်။

<small> Element က Font Size ကို သေးချင်တဲ့ အခါ အသုံးပြုပါတယ်။

<mark> Element ကို ကိုယ်က Mark လုပ်ပါး အမှတ်အသားလုပ်ချင်တဲ့အခါ သို့မဟုတ်
ထင်ရှားချင်တဲ့အခါ သုံးပါတယ်။

<sup> Element ကို Superscript အတွက်ပါ။ ဥပမာ - 5th, $a^2 + b^2 = c^2$
အပေါ်ကို တင်ထားတဲ့ 'th' တို့ square (2) တင်တာတို့ကို ရေးချင်တဲ့အခါ အသုံးပြုပါ
တယ်။

<sub> Element ကတော့ Subscript ပါ။ ဥပမာ - H₂O , x_n + y_n တို့ရေးတဲ့အခါ
အောက်ချရေးရတဲ့ စကားလုံးတွေကို လိုချင်တဲ့အခါ သုံးပါတယ်။

<ins> Element က Insert ကို ယူထားတာပါ။ သူ့ကိုတော့ Document ထဲကို အသစ် ထည့်လိုက်တဲ့ စာသားအနေနဲ့ သုံးဖို့အတွက်ရည်ရွယ်ထားတာပါ။ သို့ပေါ်မယ် Underline တားထားတဲ့ ပုံစံနဲ့သာ Browser က ပြုမှာပါ။

<u> Element က တော့ Underline တားချင်တဲ့အခါနဲ့လို့ရပါတယ်။

 Element က စာကြောင်းတွေကို အလယ်ကနေ မျဉ်းတစ်ကြောင်းဖျက်ထားတဲ့ ပဲနဲ့ မြင်ချင်တဲ့ အခါနဲ့ပါတယ်။ Delete ကို ကိုယ်စားပြုပါတယ်။

<code> - Code တွေကို HTML မှာ ဖော်ပြချင်တဲ့အခါ ဒါ Element ကိုသုံးနိုင်ပါတယ်။

<pre> - တစ်ကြောင်းထက်ပိုများတဲ့ အခါဆိုရင်တော့ <pre> Element ကို အသုံးပြနိုင်ပါတယ် သူက Space တွေကိုပါ Ignore မလုပ်တာကြောင့် ကိုယ်ရေးသလောက် သူ ကတော့ ဖော်ပြပေးသွားမှာဖြစ်ပါတယ်။

<blockquote> - အဆိုစကားတွေ အကိုးအကားတွေအတွက် အသုံးပြုကြပါတယ်။

Gist Code ကို အောက်ကလင့်ကနေ သွားရောက်ကူးယူနိုင်ပါတယ်။

<https://gist.github.com/lwinmoepaing/35c38e47a232a65762887f5d2044259f>

Codepen ကနေလည်း သွားရောက်လေ့လာနိုင်ပါတယ်။

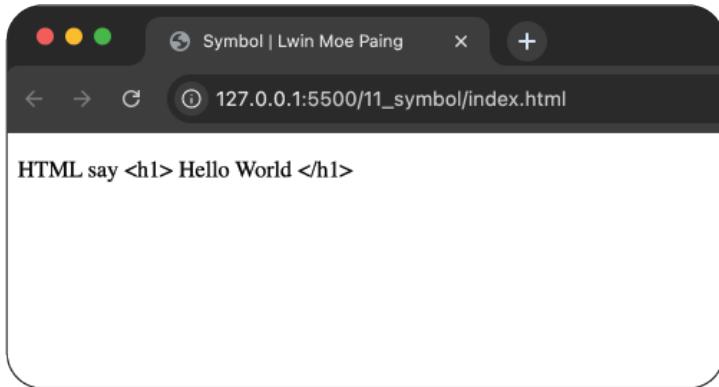
<https://codepen.io/lwinmoepaing/pen/gOQWBBJ>

HTML Symbols များလေ့လာခြင်း

<”နဲ့”> က တစ်စိတ်တစ်ပိုင်းအနေနဲ့ HTML နားလည်တယ့် Element တွေရေးတဲ့ နေရာမှာသုံးပါတယ်။ Browser က အဲဒီစာလုံးတွေတာနဲ့ Element ရေးတော့မယ်ထင် ပြီး ဘာသာပြနိပါတယ်။ အဲတော့ Browser ပေါ်မှာ စာသားအနေနဲ့ Less Than Sign ဖြစ်တဲ့ “<” တို့ Greater “>” တို့ရေးချင်တယ်အခါ ဘယ်လို့ရေးမလဲလို့ဆိုတဲ့အခါ HTML Symbol တွေသုံးပြီး ရေးကြရပါတယ်။ အောက်က Code ကို စာဖတ်သူတို့ Browser ပေါ်မှာ Run ကြည့်ပါ။

```
<p>HTML say &lt;h1&gt; Hello World &lt;/h1&gt;</p>
```

Browser ပေါ်မှာအောက်က ပုံအတိုင်း ပေါ်နေသည်ကို တွေ့ရပါလီဖို့မယ်။



အဲတော့ ထူးခြားချက်အနေနဲ့ “<” နဲ့ “>”” ကိုရေးသားချက်က ထူးခြားနေတာကို တွေ့ပြီဖြစ်ပါတယ်။ HTML Symbol တွေကို ရေးတဲ့အခါမှာ အရှေ့စာလုံးက “&” နဲ့စွဲ့ဗြို့ဗြို့ အနောက်စာလုံးက “;” နဲ့ အဆုံးသတ်ပါတယ်။ အခြား အသုံးများတဲ့ Symbol များကို ဖော်ပြု ပေးလိုက်ပါမယ်။

| | | |
|-------------|--------------|--------------|
| © - © | < - < | » - » |
| ¢ - ¢ | > - > | £ - £ |
| ® - ® | ≈ - ≈ | § - § |
| € - € | ≠ - ≠ | – - — |
| ™ - ™ | ½ - ½ | • - • |
| ≤ - ≤ | ° - ° | † - † |
| ≥ - ≥ | « - « | ‡ - ‡ |

အခုခံ Copyright ကို ရေးဖို့အတွက် © ကို Symbol နဲ့ရေးနိုင်ပါပြီ။

အောက် Gist Code လင့်ခဲ့မှာ ကူးယူပြီး Browser ပေါ်တွင်စမ်းနိုင်ပါသည်။

<https://gist.github.com/lwinmoepaing/bee0a0989757410106d8aa242dad3ce2>

Codepen ကိုအောက်တွင်ပေးထားပါသည်။
<https://codepen.io/lwinmoepaing/pen/OJamavR>

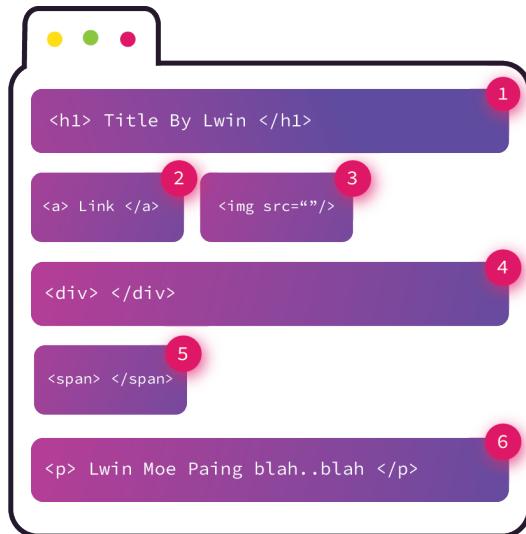
Generic Elements များဖြစ်တဲ့ <div>

သုံးခို့ဖူးတဲ့ Element တွေထဲမှာ ...

Block Elements ကို စအသုံးပြုတဲ့အခါ နောက်တစ်လိုင်းကိုကျသွားတာကိုမြင်ခဲ့ရပါတယ် သူတို့က Parent Container မှာရှိသလောက် အကျယ် (Width) ကိုယူသွားတာပါ။

Inline Elements တွေကျတော့ ဘေးကပ်ရပ်မှာပေါ်ပြီး ကိုယ့်ရှိသလောက် အကျယ်ကိုပဲယူတာလို့ပြောခဲ့ပါတယ် နောက်တလိုင်းဆင်းချင်တဲ့အခါ
 element ကို သုံးပြီး ဆင်းခဲ့ကြတာကို မှတ်မိမှာပါ။

ဆိုတော့ ကျွန်တော်တို့ အောက်က ပုံလေးနဲ့ ကြည့်ပြီး ဘယ်နံပါတ်ကတော့ Block Element ဖြစ်ပြီး ဘယ်နံပါတ်ကတော့ Inline Element ဖြစ်ကြောင်း ရွေးချယ်ကြည့်ကြမယ်။ အောက်က စာပိုဒ်မှာ အဖော်တိုက်ခင် ကိုယ်တိုင် ရွေးပြီးအရင်ဖြစ်ခေါ်ချင်ပါတယ်ပျော်။



ဟုတ်ပါပြီ ကျွန်တော်နဲ့ အတူတူအဖော်တိုက်ရအောင်

နံပါတ် (၁) <h1> Element က Block Element ပါ သူရဲ့ Parent Container က Screen တစ်ခုလုံးဖြစ်ပြီး အကျယ် (Width) ကိုအပြည့်ယူသွားပါတယ်။

နံပါတ် (၂) <a> Element က တော့ Inline Element ပါ အကျယ်ကလည်း သူရှိသလောက်ပဲယူပါတယ်။

နံပါတ် (၃) လည်း Inline Element ပါဝဲ။

နံပါတ် (၄) <div> Element က Block Element စသုံးကတည်းက အောက်တစ်လိုင်းလိုင်းအသစ်က စပါတယ် ပြီးတော့ အကျယ်လည်းအပြည့်ယူပါတယ်။

နံပါတ် (၅) Element ကတေသာ Inline Element ပါ သူရှိသလောက်ပဲ အကျယ်
ယူထားလိုပါ။

နံပါတ် (၆) <p> Element ကတေသာ Block Element ပါ။

ဒီလောက်ဆို စာဖတ်သူတို့ Inline Element နဲ့ Block Element ကို ကွဲကွဲပြားပြားသိသွား
ပြီလို ယူဆပါတယ်ခင်ပျော်။

<div> Element ကတေသာ Group အနေနဲ့ သို့မဟုတ် အခြား Elements တွေကို Organized ဖြစ်ဖို့အတွက် သုံးပါတယ်။ သူက Box တစ်ခုနဲ့ အလားတူပါတယ် သူ့အထူး
အခြား elements တွေအများကြီးရှိနိုင်တယ် image ပုံတွေ၊ စာသားတွေ အများကြီးပါ
နိုင်ပါတယ်။ တကယ်လောက်တွေ၊ Real Project တွေမှာ တွေ့ရအများဆုံး Element တစ်
ခုပါ။

ဘာလိုအသုံးများလည်းဆိုရင် ဥပမာ -

<h1> ဆိုတာ ခေါင်းစဉ်အတွက် သုံးဖို့ရည်ရွယ်ထားတဲ့ Element အမျိုးအစားပါ။

 ဆိုတာက စာပိုဒ်တွေအတွက်ဖြစ်ပြီး။

 က ပုံအတွက်ပါ။

ဒါဆိုရင် ခေါင်းစဉ်တွေကော်၊ စာပိုဒ်တွေကော်၊ ရုပ်ပုံတွေပါ ပါတဲ့ Card Element တစ်
ခုတည်ဆောက်မယ်ဆိုရင် အဲလို ရည်ရွယ်ချက်ရှိတဲ့ Card Element က မရှိပါဘူး
ဒီလိုအခြားနေရာမျိုးမှာ

<div> Element ကို Container သဖိယ် သူ့ကို အပ်ပြီးသုံးလေးရှိပါတယ်။

ဆိုလိုချင်တာက <div> Element က Container သဘောမျိုးဖြစ်သွားပြီး သူ့အထူး

<h1><p> သုံးမျိုးလုံးကို ထည့်ရေးလိုက်ပါတယ်။

 Element ကလည်း <div> သဘောမျိုးပါပဲ ရည်ရွယ်ချက်ရှိတဲ့ Element မဟုတ်
ပေးမယ်၊ ကျွန်ုတ်တို့ ခုကာကခွဲခြားခဲ့တဲ့ Block (or) Inline Element အမျိုး
တဲ့ဖြစ်ပါတယ်။ <div> Element ကတေသာ Block Element ပါ။

 Element ကိုအဓိကအားဖြင့် Text တွေကို Decoration လုပ်တဲ့အခါမျိုးမှာ Css
နဲ့သုံးလေးရှိပါတယ်။ ဥပမာ Element ထဲက စာသားကို အနီရောင်ပြောင်
ပေးတာတို့ အသေးစိတ်လိုအပ်လာတဲ့အခါ Inline Element အနေနဲ့ အပ်ပါးအသုံးပြုလေ့
ရှိပါတယ်။

ဒီလောက်ဆို အခြေခံအားဖြင့်တော့ <div> ရယ် ရယ်ကို နားလည်သလောက်ရှိ
သွားပြီလိုယူဆပါတယ်။ CSS အခန်းမှာ ပိုပြီးတော့ အသုံးပြုမှာဖြစ်တဲ့အတွက်ကြောင့်
ဆက်လေ့လာသွားပါမယ်။

Comment in HTML <!-- -->

ဒီ <!-- --> Element က အရမ်းအရေးပါပါတယ်။ ကိုယ်မှတ်ချင်တဲ့ စာသားတွေကို
မှတ်ထားလို့ရတဲ့ Element ပါ။ Browser က Comment ပိတ်ထားတဲ့ Content တွေကို
ဖော်ပြုမှုမဟုတ်ပါဘူး။ အောက်က ပုံလေးကိုကြည့်လိုက်ပါ။

```

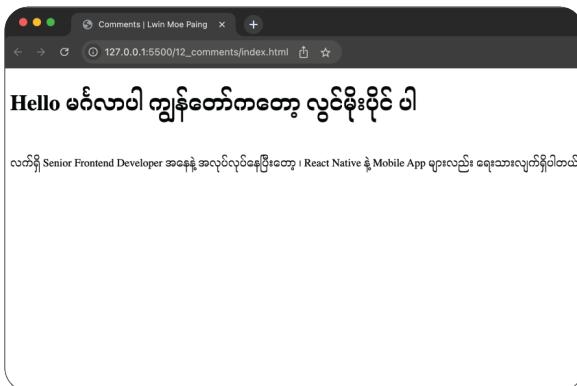
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<title>Comments | Lwin Moe Paing</title>
</head>
<body>
    <!-- Start of Introduction -->
    <h1>Hello မင်္ဂလာပါ ကျွန်တော်ကတော့ လွှဲပို့ပိုင် ပါ</h1>
    <!-- End of Introduction -->

    <!-- Start of About Me -->
    <p>
        လင်္ဂလာ Senior Frontend Developer အင်္ဂ အလုပ်လုပ်ငန်းတော့ React Native  

        နဲ့ Mobile App များလုပ် ရရှိသော်လည်းကောင်းမူပါတယ်
    </p>
    <!-- End of About Me -->
</body>
</html>

```

အညွှန်း (၁) အရ <!-- Start of Introduction --> နဲ့ <!-- End of Introduction -->
ကိုတွေ့ရမှုဖြစ်ပြီး။ အညွှန်း (၂) အရ <!-- Start of About Me --> နဲ့ <!-- End of
About Me --> ကိုတွေ့ရမှုပါ။ Browser မှာ ပေါ်လာမယ့်ပုံကိုအောက်မှာကြည့်လိုက်ပါ။



ရေးသားတွေတဲ့ Comment Element တွေကို Browser က ပြန်လည်ဖော်ပြပေးမှုမဟုတ်
ပါဘူး။ HTML code တွေထဲမှာကျွန်တော်တို့က Comment ရေးသားထားတဲ့အကျင့်
ရှိတာကောင်းပါတယ်။ တကယ် လက်တွေ့ Project တွေမှာ Code လိုင်းပေါင်းက
ထောင်ချိရှိရှိပို့ပါတယ်။ နောက်တပါ ပြန်ပြင်တဲ့အပါ ကျွန်တော်တို့က ဒီ Element က
ဘာအတွက်ရေးသားတာပါဆိုပြီး Comment ရေးထားမယ့်ဆိုရင် အဗြား Developer
ကလည်း နားလည်ပြီး ပြင်ဆင်ရလွယ်ကူနိုင်ပါတယ်။

HTML5 Layout Elements များ

HTML5 Layout Element များက Webpage တစ်ခုရဲ့ မတူညီတဲ့ အပိုင်းတွေကို Structure ကျကျဖော်ပြသော Element များဖြစ်ပါတယ်။ လေ့လာခဲ့သော <div> Element ကို ပြန်တွေးကြည့်လိုက်ပါ သူက Container သဘောမျိုး အမြား Element များကို အပ်ပြီး ရေးလေ့ရှိပါတယ်။ <div> တိုးပဲသုံးနေမယ့်အတာ၊ Layout Element များကို သုံးခြင်း ဖြင့် ပိုမိုရှင်းလင်းသော Code ဖြစ်သွားစေပါတယ်။ ဂျွန်တော်တို့ အောက်က ပုံလေးကို လေ့လာကြည့်ရအောင်။

< body >



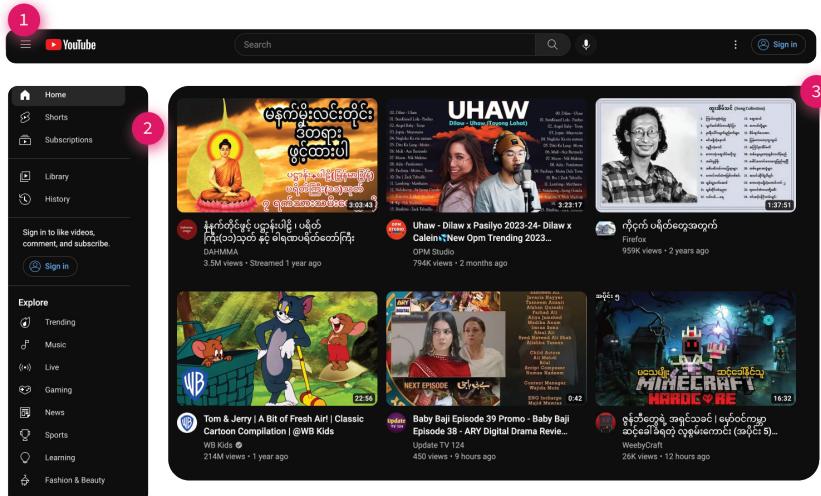
< /body >

ပုံကို ကြည့်ပြီး HTML5 Layout Element တွေကိုလေ့လာကြည့်ရအောင်။

- <header> ဆိတာခေါင်းစဉ်ပိုင်းတွေရေးတာဖြစ်ပြီး <footer> ကလည်း Web Page ရဲ့ အောက်ခြေမှာရေးလေ့ရှိပါတယ် Contact Info တွေ ဖုန်းနံပါတ်တွေနဲ့ CopyRight တွေ အမြဲတွေ့ရလေ့ရှိတာကို စာဖတ်သူတို့သိကြမှာပါ။
- <nav> က Navigation ဆိတ္တဲ့ လမ်းညွှန် Element တစ်ခုပါ သူ့အတွင်းမှာဆို Anchor Link တွေအများဆုံးပါဝင်ပါတယ်။ ဘယ် Page တွေရှိတယ်သွားနိုင်တယ်ဆိုတာမျိုးပါ။
- <aside> က တော့ ဆပ်စပ်နေသော Content တွေပြို့ပါ သတင်း Web Page တွေမှာ ဆိုရင် အလားတူ သတင်းတွေကို ဆက်ကြည့်ချင်ရင် သေးမှာမြင်ရသလို့ပေါ့။

- <article> Element ကတော့ သူရဲ့ Element Name အတိုင်းပဲ Articles တွေရေးသား နိုင်ဖို့ပါ။
- <section> Element က ပုံထဲမှာတော့မပါပါဘူး စဉ်းစားကြည့်ရအောင် Home Page တစ်ခုမှာ Section တွေအများကြီးပါနိုင်ပါတယ်။ ဥပမာ - ယနေ့ သတင်းများ၊ Popular သတင်းများ၊ သတင်းအဟောင်းများဆိုပြီး <section> Element များမြင့် (၃) ပိုင်းဆွဲပြီး ရေးနိုင်ပါတယ်။
- <figure> Element က ပုံထဲမှာထည့်သွင်းမဖော်ပြထားပါဘူး သူက ပုံပါမယ် ပုံရဲ့ Caption ပါမယ်ဆိုပို့။ ပုံအတွက် Element အသုံးပြုပြီး Caption အတွက်တော့ <figcaption> ဆိုပြီးအသုံးပြုနိုင်ပါတယ်။
- <main> Element ကလည်း အဓိကအကျဉ်း Content တွေကိုဖော်ပြချင်ရင် အသုံးပြုလေ့ရှိပါတယ်။

ကျွန်တော်တို့ ပိုရှင်းသွားအောင် အောက်က ပုံလေးကို လေ့လာကြည့်ရအောင်



ဒါကတော့ စာဖတ်သူတို့ သိကြပါမယ့် Youtube Site ပဲဖြစ်ပါတယ်။

- နံပါတ် (၁) က <header> Element ဖြစ်ပါတယ် သူက ခေါင်းစီးပိုင်းဖြစ်ပြီး Logo တို့ Search တိုပါပါတယ်။
- နံပါတ် (၂) က တော့ <nav> Element ဖြစ်ပါတယ် သူထဲမှာ သက်ဆိုင်ရာ Link တွေကို ညွှန်းထားတဲ့ လမ်းညွှန်းလိုအောင်ရှုက်နေလိုပါ။
- နံပါတ် (၃) က တော့ Section Element ဖြစ်ပါတယ် အထဲမှာ အခြား Content တွေ ပုံတွေ အများကြီးပါမှာပါ။
- ဒီလောက်ဆို Layout Element တွေကိုဘယ်လိုသုံးလဲ သိသွားပြီလို့ ယူဆပါတယ်။

အခု HTML အပိုင်းကို ကျွန်တော်ဒီမှာပဲ ရပ်လိုက်ပါတော့မယ်။ ဒါအကုန်ပဲလားဆိုတော့ မဟုတ်ပါဘူး။ HTML ထဲမှာ <canvas> လို ပုံဆွဲတဲ့အခါ Animation တွေလုပ်တဲ့အခါ သုံးတဲ့ Element တွေကျွန်ပါသေးတယ်။

Video တွေပြချင်ရင်သုံးတဲ့ <video> Element နဲ့ Audio တွေအသုံးပြုတဲ့ အခါ သုံးတဲ့ <audio> Element တွေလည်းရှိပါသေးတယ်။

အသုံးအများဆုံး Element တွေကို လေ့လာခဲ့ပြီးဖြစ်ပါတယ်။

အခုထိ သေသေချာချာ Practical တွေရော လေ့ကျင့်ခဲ့မယ်ဆိုရင် စာဖတ်သူက လုံးဝ HTML မှာ Beginner မဟုတ်တော့တာ ကြိမ်းသေပါပဲ။

နောက်စာမျက်နှာတွေမှာ CSS အကြောင်းအရာများကို ဆက်လက် လေ့လာသွားကြပါ မယ်။



HTML

CSS

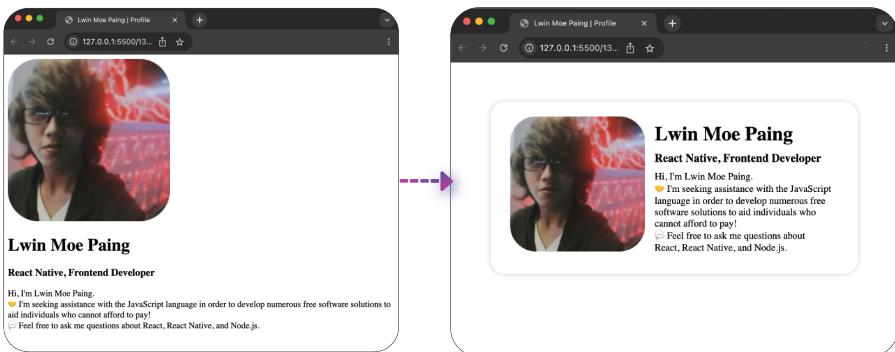
မိန်းကလေး မင်းသာ Web Dev အကြောင်းနားလည်မယ်ဆိုရင်
မင်းဟာ ကိုယ့်ဘဝရဲ့ CSS ပါပဲ။ မင်းမရှိရင်
ကိုယ့်ဘဝကြီးက မလုပတော့ဘူးလေ

CSS for Dummies

CSS က ဘာလုပ်လဆိုရင် HTML ရဲ့ Browser ပေါ်မှာ ပေါ်လာမယ့် Apperance တွေနဲ့ Style တွေကို သတ်မှတ်ပေးတဲ့အလုပ်လုပ်ပါတယ်။

CSS ရဲ့ အရှည်ကတော့ Cascading Style Sheet ဖြစ်ပါတယ်။ CSS နဲ့ ဘာတွေလုပ်နိုင်လဆိုရင် Font တွေဖြစ်တဲ့ စာသားတွေ၊ Color အရောင်တွေ၊ Positioning ဖြစ်တယ့်၊ Element Layout တွေကို Rule တွေသတ်မှတ်ပြီး ပြောင်းလဲပေးနိုင်ပါတယ်။ ဒါအပြင် Animation တွေပါ CSS က Control လုပ်နိုင်စွမ်းရှုပါတယ်။

HTML for Dummies အခန်းမှာတုန်းက ပထားဆုံး စတွေ့ခဲ့သော ကျွန်ုတ် Profile Site လေးကို လေ့လာခဲ့ကြပါတယ်။ အထဲက HTML နဲ့ ရေးထားသော Element များ ကို လုံးဝကြောင်းခဲ့ခြင်းမပြုပဲ CSS ပေါင်းထည့်လိုက်တာနဲ့ ပြောင်းလဲသွားတာကို အောက်က ပုံ (၂) ပုံကို ဖြင့် နှင့်ယူလိုကြည့်ပါ။



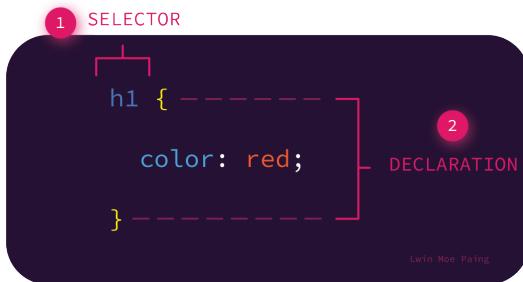
ပုံ (၁) က ရှိုးရှိုး HTML တွေနဲ့ပေးသားထားတဲ့ Page ဖြစ်ပါတယ်။ CSS လေးနဲ့ ကျွန်ုတ်တို့ ပေါင်းထည့်လိုက်တဲ့အခါ Screen ရဲ့ အလယ်ကို ရောက်သွားမှုဖြစ်ပြီး Shadow အရိပ်လေးနဲ့ ကတိဒ် ပုံစံလေးပါပြောင်းလည်းပေးလိုက်ပါတယ် ပြီးတော့ ပုံ ရဲ့ ညာဘက်တစ်ဖက်တစ်ချက်ဆိုတဲ့ Heading Element နဲ့ Paragraph တစိုက် Element တွေက ရောက်ရှိသွားမှာကို မြင်တွေရမှာပါ။ ဒီလောက်ဆို HTML ရဲ့ ဒီဇိုင်းတွေကို CSS ကပြောင်းလဲပေးနိုင်တာကို အကြမ်းဖျဉ်းအားဖြင့်သိရှိသွားပြီလို့ ယူဆပါတယ်။

ဒါကို Codepen လင့်ကိုအောက်မှာဖော်ပြပေးထားပါတယ် စာဖတ်သူတို့က သွားရောက်လေ့လာနိုင်ပါတယ်။

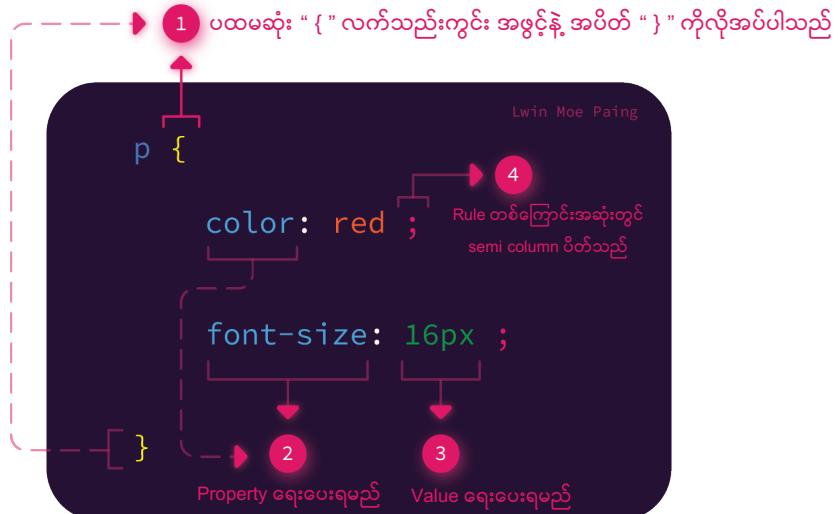
<https://codepen.io/lwinmoeapaing/pen/JjeNQyw>

CSS ရေးသားဖွဲ့စည်းပုံကို လေ့လာခြင်း

HTML Element တွေကို Style ဒီဇိုင်းပုံဖော်ပို့အတွက် CSS ကို Rule တွေသတ်မှတ်ပြီး အသုံးပြုကြပါတယ်။ Rule တွေသတ်မှတ်ရာမှာ အပိုင်း (J) ဖိုင်း ပါဝင်တယ်။ တစ်ပိုင်းက ရွေးချယ်မှု Selector ဖြစ်ပြီး။ နောက် တစ်ပိုင်းကို Declaration (ကြော်ငြာချက်သတ်မှတ်ခြင်း) လို့ခေါ်ပါတယ်။



အညွှန်း နံပါတ် (၁) ကိုကြည့်လိုက်ပါ သူက h1 ဆိုပြီး Selector ကိုမြင်ရမှာပါ။ အညွှန်း နံပါတ် (၂) ကိုကြည့်ရင် လက်သည်းကွင်း အစ က နေ လက်သည်းကွင်း အဆုံးထိ ကြားထဲက ဟာကို Declaration လို့ခေါ်ပါတယ်။ အမိပ္ပလားက CSS ဖြင့် <h1> Elements များကို လက်သည်းကွင်း အထဲတွေ့ ကြော်ငြာထားသော "color: red" Rule အရ အရောင်တွေကို အနီးရောင်အဖြစ် သတ်မှတ်ပါသည် ဆိုတဲ့ အမိပ္ပလား ဖြစ်ပါတယ်။ ဒါမူဘေးမှာ အမြတ်မြတ်မှုများ မျှတော်းဆောင်ရွက်နေတဲ့ အမြတ်မြတ်မှုများ မျှတော်းဆောင်ရွက်နေတဲ့ ...

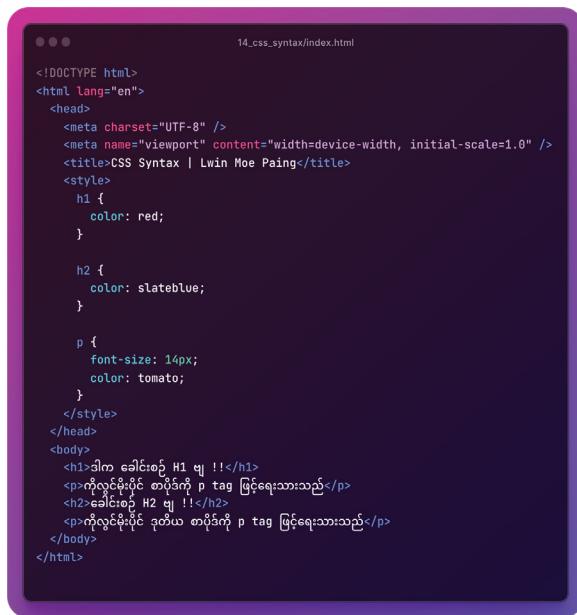


Declaration ကိုလေ့လာရတွင် အထဲတွင် Rule တစ်ခုထက်မက သတ်မှတ်နိုင်သည်ကို တွေ့ရပါမယ်။ ဥပမာပေးထားသောပုံတဲ့တွင် -

“color: red;” ဆိုသည်မှာ <p> Element အတွက် အနီရောင်စာသားသတ်မှတ်သည်။ “font-size: 16px;” ဆိုသည်မှာ <p> Element အတွက် Font Size 16px သတ်မှတ်သည်။ ဆိုပြီး Rule နှစ်ခုသတ်မှတ်လိုက်ဖော်ပြုနိုင်သူများ တွေ့ရပါမယ်။

အခုခို CSS ရဲ့ Syntax ရေးထုံးကို ယေဘုယျအားဖြင့် ကျွန်တော်တို့လေ့လာပြီးပါပြီ။

ပထမဆုံး ကျွန်တော်တို့ Folder ကို “14_css_syntax” လိုပဲ ဖိုလ်ခါ နာမည်လေး ပေးလိုက်မယ်၊ အထဲမှာ “index.html” HTML file တစ်ခုတည်ဆောက်ပြီးရင် အောက်က Code အတိုင်း ကျွန်တော်တို့ Browser မှာသွား Run ကြည့်ရအောင် ...



```

14.css_syntax/index.html

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>CSS Syntax | Lwin Moe Paing</title>
    <style>
        h1 {
            color: red;
        }

        h2 {
            color: slateblue;
        }

        p {
            font-size: 14px;
            color: tomato;
        }
    </style>
</head>
<body>
    <h1>ခါော ခေါင်စဉ် H1 မှု !!</h1>
    <p>ဂိုလ်မြိုင်ပိုင် အုပ်စုရီ ပု tag မြင်စေသားသည်</p>
    <h2>ခေါင်စဉ် H2 မှု !!</h2>
    <p>ဂိုလ်မြိုင်ပိုင် အုပ်စုရီ ပု tag မြင်စေသားသည်</p>
</body>
</html>

```

Gist Code ကိုအောက်က Link ကဖန်တီးပူးပါနိုင်ပါသည်။

<https://gist.github.com/lwinmoepaing/2a0119e34fdeb9906227b0173d4b8c96>

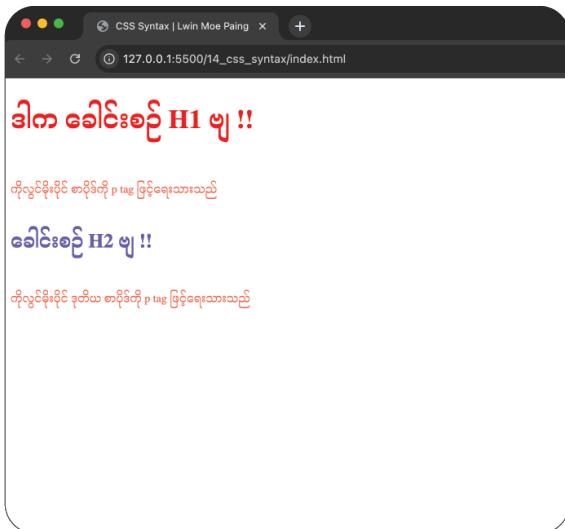
Codepen ကိုအောက်က လင့်ခြားဖြင့် Online ပေါ်တွင် တိုက်ရှိက် စမ်းနိုင်ပါသည်။

<https://codepen.io/lwinmoepaing/pen/NWEgKrm>

CSS ကိုအသုံးပြုရာတွင် နည်းလမ်း (၃) မျိုးဖြင့်အသုံးပြုနိုင်ပါတယ်။

External Css | Internal Css ရယ် Inline Css ဆိုပြီးရှုပါတယ် ကျွန်တော် နောက်
တစ်မျက်နှာကျရင် ရင်းပြသွားပါမယ် အခု Example Code ထဲမှာ <head> Element
အတွင်း <style> Element ကိုပြုလုပ်ပြီး CSS Code များရေးသားခဲ့ကြသည်ကို တွေ့ရ
ပါမယ်။ ထို့ပုံး အသုံးပြုနည်းက Internal Css နည်းဖြင့် အသုံးပြုနည်းဖြင့်ပါတယ်။

Example Code အရ အောက်က ပုံအတိုင်းရလာပါမည်။



အောက်က ကျွန်တော်တို့<style> Element ထဲမှာရေးသားခဲ့တာတွေနဲ့ တိုက်ကြည့်မယ်

Lwin Moe Paing

```
h1 {-----}
    color: red;
-----} Selector <h1> element ကို
          အနိရောင် စာသားသတ်မှတ်မည်
```

```
h2 {-----}
    color: slateblue;
-----} Selector <h2> element ကို Slate Blue
          အရောင် စာသားသတ်မှတ်မည်
```

```
p {-----}
    font-size: 14px;
    color: tomato;
-----} Selector <p> element ကို
          Font Size (စကားလုံးဆိုင်) 14 px နှင့်
          စာသားအရောင် Tomato အရောင်
          သတ်မှတ်မည်
```

သတ်မှတ်ထားသော Style အတိုင်း Browser ပေါ်တွင်ပေါ်နေသော ပုံနှင့်တိုက်ကြည့်ပါ
စာဖတ်သူတို့ ပိုမို ရင်းလင်းစွာနားလည်သွားပါလိမ့်မည်။

CSS အကြောင်းအရာများ

CSS ကို Inline Style အသုံးပြုခြင်း

CSS ကိုအသုံးပြုရာတွင် နည်းလမ်း (၃) မျိုးဖြင့်အသုံးပြုနိုင်ပါတယ်။

External Css | Internal Css ရှုလို အသုံးပြုနိုင်ပါတယ်။ ပထမဆုံးဖော်ပြမှာ ကတေသာ အသုံးပြုကြည့်ပါ။

အရင်ဆုံး ကျွန်ုတ်က New Folder (ဖိုလ်ဒါအသစ်) အနေနဲ့

“15_css_using_method” လို ပေးလိုက်ပါတယ်။ အထဲမှာ “inline.html” ဆိုပြီး HTML file တည်ဆောက်လိုက်ပါမယ်။ ပြီးရင် အောက်က Code အတိုင်းယူပြီး စာဖတ်သူတို့ Browser မှာအသုံးပြုကြည့်ပါ။

Gist Code ကို အောက်က လင့်ခဲ့မှ ကူးယူနိုင်ပါသည်။

<https://gist.github.com/lwinmoepaing/174639e71c57491230256d58a92d8d53>

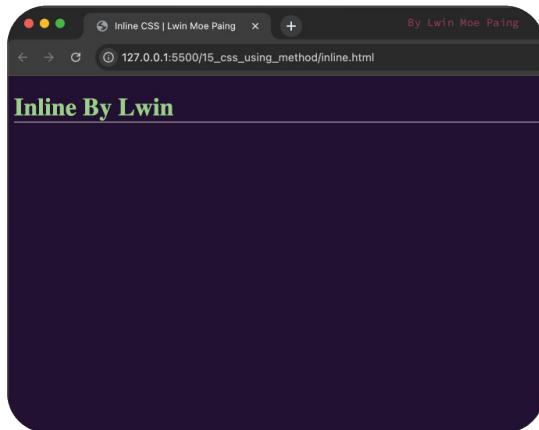
Code Pen လင်ခဲ့ကို အောက်က လင့်ခဲ့မှ ဝင်စမ်းနိုင်သည်။

<https://codepen.io/lwinmoepaing/pen/wvQevOd>



ပြီးနောက် စာဖတ်သူတို့ Browser မှာ Run ကြည့်လိုက်တဲ့ပါ။

အောက်က အတိုင်း ပေါ်နေသည်ကို တွေ့ရပါမယ် ...



Inline ရေးနည်းက ရှင်းပါတယ် HTML Element အတွင်းမှာ Attribute style ထဲ CSS Rule တွေကို ထည့်ရေးလိုက်ခြင်းဖြစ်ပါတယ်။ အောက်က ပုံကိုတစ်ချက်လေးလာကြည့်ရအောင် ...

```
<body style="background-color: #241133">
    ...
</body>
```

Property Value

By Lwin Moe Paing

1

အထဲမှာ background-color ဆိုတဲ့ Property နဲ့ သူ့ရဲ့ Value အနေနဲ့ "#241133" ဆိုတာကိုတွေ့ပါလိမ့်မယ်။ Hex Code နဲ့ ဖော်ပြထားတာဖြစ်ပါတယ် ကျွန်ုတ် Color အနီးရောက်ရင် Color အကြောင်းနဲ့ Unit တွေအကြောင်းအသေးစိတ်ရှင်းပြုပါမယ်။ <body> Element ရဲ့အရောင်ကို hex code အတိုင်း ပြောင်းထားကြောင်းသတ်မှတ်ထားတာပါ။

```
<h1 style="color: lightgreen ; border-bottom: 1px solid rgb(255, 255, 255)">
    ...
</h1>
```

Divided by ";"

By Lwin Moe Paing

2

<h1> Element အတွက် Color ကို lightgreen အနေနဲ့ထားတာကို နားလည်မှာပါ။ သတိထားရမှာက နောက်ထပ် Rule တစ်ခုအတွက် ကြားထဲမှာ ";" Semi Column ကို ထည့်ပေးရမှာကို သတိထားပေးပါ။ ခုတိယ Rule တစ်ခုက အောက်ပျဉ်လိုင်းလေးတားဖို့အတွက် "border-bottom" Attribute ထဲကို 1px ရှိတဲ့ Solid Style နဲ့ အရောင်က RGB Color တန်ဖိုးကို ထည့်ပေးလိုက်တာဖြစ်ပါတယ်။ RGB ဆိုတာလည်း Color တန်ဖိုးဖြစ်ပါတယ်။ Color အနီးရောက်ရင် ရှင်းပြုပါမယ်။

အခုဆို ကျွန်တော်တိ။ Inline Style ပုံစံနဲ့ အသုံးပြုခြင်းကို နားလည်သွားပြီဖြစ်ပါတယ်။

မှတ်ချက်။ ။ Inline Style ကို Real Project တွေမှာ သုံးခြင်းကို အတတ်နိုင်ဆုံး ရောင်ကြပါတယ်။ အရမ်းလိုအပ်တဲ့အခါမှသ သုံးပါတယ် သူက Override ပုံစံမျိုးဖြစ်သွားပါတယ်၊ ဆိုလိုချင်တာက အခြားပုံစံ External (or) Internal Style Method နဲ့သုံးထားတာမျိုးကို ထပ်အပ်ပြီး အသုံးပြုသလိုဖြစ်သွားတာပါ။ မကောင်းတဲ့အချက်တွေ ကတော့ Resuable မဖြစ်တာပါ ထပ်ခါထပ်ခါ Style တစ်ခုတည်းကိုအသုံးပြုချင်တဲ့အခါ Element တိုင်းရဲ့ အတွင်းထဲမှာလိုက်ရေးနေရတာပါ။ ပြန်လည်ပြင်ဆင်ပယ်ထိုလည်း Inline Style သုံးထားတော့ Element တိုင်းကို ပြင်ဆင်ရခါက်ခဲခြင်းဖြစ်နိုင်ပါတယ်။ ကုတ်တွေလည်း ရှုပ်ထွေးပြီး များပြားနိုင်ပါတယ်။ သို့ပေမယ့် နောက်ပိုင်း ခေတ်ပေါ် Tech တွေဖြစ်တဲ့ React မှာတော့ အသုံးများပါတယ်။

CSS ကို Internal Style အသုံးပြုခြင်း

CSS ရေးသားစွဲစည်းပုံလေ့လာခြင်း ခေါင်းစဉ်မှာတုန်းက အသုံးပြုခဲ့တဲ့ `<style>` Element ကို မှတ်မိကြေးမယ်ထင်ပါတယ်။ ပုံစံက ကျွန်တော်တို့ `<head>` Element ထဲမှာ ရေးသားပြီးတော့ CSS syntax ရေးထုံးအရ Selector နဲ့ Declaration ပုံစံအတိုင်း ရေးသားရမှာဖြစ်ပါတယ်။ ခုကဗာ Inline Style အသုံးပြုခဲ့တုန်းက ပုံစံအတိုင်းကို Internal Style ဖြင့်ပြောင်းလည်းရေးသားမှာဖြစ်သောကြောင့် Browser တွင် ယခင်က Example အတိုင်း ရလဒ် Result ရရှိပါမည်။

အရင်ဆုံး “15_css_using_method” Folder (ဖိုလ်ဒါ) အထဲမှာ “internal.html” ဆိုပြီး HTML file တည်ဆောက်လိုက်ပါမယ်။ ပြီးရင် အောက်က Code အတိုင်းယူပြီး စာဖတ်သူတို့ Browser မှာအသုံးပြုကြည့်ပါ။

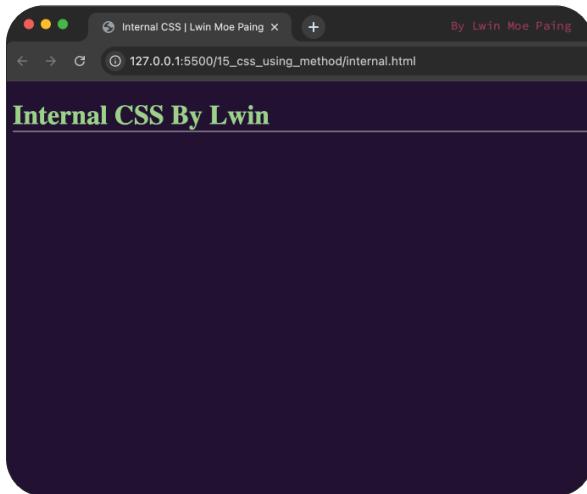
Gist Code ကို အောက်က လင့်ခဲ့မှ ကူးယူနိုင်ပါသည်။

<https://gist.github.com/lwinmoepaing/ff03385cd2fac996c90804b46639e7c2>

Codepen ကို အောက်က လင့်ခဲ့မှ ကူးယူနိုင်ပါသည်။

<https://codepen.io/lwinmoepaing/pen/VwVWYVb>

Browser ပေါ်တွင် Run လိုက်ပါက အောက်က ပုံတိုင်း ပေါ်လာပါမည်။



Internal ရေးထာက ပထမအချက် <style> Element တို့ရေးပေးရပါမယ်။ နောက်
တော့ CSS တွေကို Selector နဲ့ Declaration (Property + Value) အရ Rules တွေကို
သတ်မှတ်ပေးပြီးတော့ ရေးပေးရပါမယ် အောက်ကပုံကို ကြည့်ပေးပါ ...



ကျွန်ုတ်တို့ CSS Internal ရေးနည်းကို လေ့လာပြီးဖြစ်ပါသည်။

မှတ်ချက်။ Internal ရေးနည်းတွင် <head> Element ထဲတွင် <style> Elements
များကို ရေးမှတ်ထားလေးရှိပါသည်။

CSS ကို External Style အသုံးပြုခြင်း

အသုံးအများဆုံးရေးနည်းဖြစ်ပါတယ် External ဆိုတဲ့အတိုင်း ကျွန်တော်တို့က CSS ကို ဖိုင်တစ်ခုအနေနဲ့ ခွဲထဲတို့ပြီးရေးကြရမှာပါ။ နောက်ပိုင်း လေ့လာကျမယ့် CSS Library တွေ Tailwind Css တို့ Bootstrap တို့ကိုအသုံးပြုမယ့်အခါမှာလည်း ဒီလို External ရေးနည်းကို မြင်တွေ့ကြရမှာဖြစ်ပါတယ်။

အရင်ဆုံး “15_css_using_method” Folder (ဖိုလ်ဒါ) အတဲ့မှာ “external.html” ဆိုပြီး HTML file တည်ဆောက်လိုက်ပါမယ်။ နောက် CSS File (ဖိုင်) ကို “style.css” ဆိုပြီး တည်ဆောက်လိုက်ပါမယ်။ ဒါဆိုရင် html တစ်ဖိုင်ရယ် css ဖိုင်တစ်ဖိုင်ရယ် ရပါပြီ။

HTML နဲ့ CSS File အတူတူ ဘယ်လို အလုပ်လုပ်လဲဆိုတာကို အောက်ကပုံတဲ့က Code တွေကို လေ့လာကြည့်ရအောင် ...

external.html ဖိုင်ထဲကို အောက်က Code အတိုင်းရေးသားပါမည်

```
15_css_using_method/external.html By Lwin Moe Paing

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>External CSS | Lwin Moe Paing</title>
    <link rel="stylesheet" href="style.css" type="text/css" />
  </head>
  <body>
    <h1>External CSS By Lwin</h1>
  </body>
</html>
```

style.css ဖိုင်ထဲတွေ့က အောက်က Code အတိုင်းရေးသားပါမည်

```
15_css_using_method/style.css

body {
  background-color: #241133;
}

h1 {
  color: lightgreen;
  border-bottom: 1px solid rgb(255, 255, 255);
}
```

By Lwin Moe Paing

Browser ပေါ်တွင် Run လိုက်သောအခါ အောက်ပါ ပုံအတိုင်းမြင်ရပါမည်။



CSS ကိုဖိုင်သီးချွဲပြီး ရေးရာတွင် ချိတ်ဆက်ပုံကို သတိထားမိလိုက်ပါသလား။ ဟုတ်ပါတယ် `<head>` Element အတွင်းမှာ `<link>` ဆိုတဲ့ Element ကို အသုံးပြုလိုက်ပါတယ်။ `<link>` Element က ဘာတွေလုပ်လဲဆိုရင် External အားဖြင့်မှာရှိသော Resources တွေကို အခြေခံနေတဲ့ HTML နဲ့ချိတ်ဆက်ပေးပါတယ်။ External Resources တွေထဲမှာ CSS ဖိုင်တွေလည်းပါသလို HTML tab ဘားမှာပေါ်ချင်တဲ့ Icon အစရိုသည်တို့ ကို Link ချိတ်ပေးလို့ရပါတယ်။ Attribute (၃) ခုရှိပါတယ် `type` ကတေသ့ ဖိုင်ရဲ့၊ အမျိုးအစားကို သတ်မှတ်ပေးရတာပါ "text/css" ဆိုတာ CSS ဖိုင်အမျိုးအစားဖြစ်ပါတယ်လို့ သတ်မှတ်လိုက်တာပါ။ "rel" Attribute ကတေသ့ HTML နဲ့ ချိတ်ဆက်မယ့် ဖိုင်က ဘယ်လို Relationship ရှိကြောင်းသတ်မှတ်ပေးရတာပေါ့ ဒီနေရာမှာ "stylesheet" အနေနဲ့ relative ဖြစ်တယ်လို့ သက်ရောက်ပါတယ်။ "href" Attribute ကတေသ့ ဖိုင်ရဲ့ တည်နေရာ URL ထို့မဟုတ် Path လမ်းကြောင်းကို သတ်မှတ်ပေးရတာပါ။ အောက်ကပုံကို တစ်ချက်ကြည့်ပြီး လေ့လာကြည့်ကြပါစို့ ...



အလွတ်မှတ်မိစရာတော့မလိုပါဘူး အသုံးများသွားတဲ့အခါ အလိုလိုမှတ်မိသွားပါလိမ့်မယ် Code Editor အရ VS Code မှာဆိုရင် Emmet Abbreviation နဲ့ `link:css` ရှိကဲလိုက်ပါက တခါတည်းတန် Auto Link Element တည်ဆောက်ပေးသွားပါလိမ့်မည်။

အခုဆို CSS ရဲ့ အသုံးပြုပုံ (၃) မျိုးစလုံးကိုလေ့လာပြီးဖြစ်ပါတယ်။ ခုကာ Code တွေကို Gist နဲ့ Codepen လင်ခဲ့ကို အောက်တွင် ဖော်ပြထားပါတယ်။

<https://gist.github.com/lwinmoepaing/26077252d4f1d390556184f10349f1af>

<https://gist.github.com/lwinmoepaing/1a4472c91155a83a9e5ed07c88a7def8>

<https://codepen.io/lwinmoepaing/pen/dyQRYMw>

CSS Selector ဆိုတာ

CSS မှာ Rules တွေသတ်မှတ်စိုး အရင်ဆုံး Selector ရေးပြီးနောက် Declaration (ကြိုးကြောက်) များသတ်မှတ်ရတယ်ဆိုတာ လေ့လာပြီးဖြစ်ပါတယ်။

ဥပမာ -

h1 { color: red } မှာဆိုရင် h1 က Selector ပါ။

အလားတူ h1 , h2, h3 { color: red } ဆိုရင် ထို Elements (၃) မျိုးလုံးကို အနီးရောင် အဖြစ်သတ်မှတ်လိုက်တာပါ။ ထူးခြားချက်အရ “,” comma ဆံသုံးပြီးတော့ တစ်ခုထက် မက ရွှေးချယ်ပေးနိုင်ပါတယ်။

အသုံးအများဆုံး Selector တွေကို စာမျက်နှာ နောက်တစ်မျက်နှာတွင် Selector Table ပေားကို ကြည့်ပြီး လေ့လာကြည့်ပါမယ်။

| ချွေးချယ်မှု - Selector | အဓိပ္ပာယ် | ဥပမာ |
|-------------------------|---|---|
| UNIVERSAL SELECTOR | Page မှတ်၍သော Element အားလုံးကို သက်ဆိုင်သည် Applied လုပ်ခြင်း။ | * {} Star နဲ့ “{}” အတွင်းရှိ CSS rules တွေက Element အားလုံးနဲ့သက်ဆိုင်ပါတယ်။ |
| TYPE SELECTOR | Element Type နာမည်နဲ့ တူတဲ့ Element တွေအပေါ် Applied လုပ်ခြင်း။ | p, span {} Target တွေဖြစ်တဲ့ p, span အစရှိတဲ့ Element တွေနဲ့ သက်ဆိုင်ပါတယ်။ |
| ID SELECTOR | Id Attribute တန်ဖို့နဲ့ တူတဲ့ Element အပေါ် Applied လုပ်ခြင်း။ | #top {} Target တွေ id Attribute ဖြစ်တဲ့ Element နဲ့ သက်ဆိုင်ပါတယ်။ eg - <div id="top"> ... </div> |
| CLASS SELECTOR | Class Selector “.” (FullStop) အနောက်ရှိ နာမည်နဲ့ တူသော Class Elements များအပေါ် Applied လုပ်ခြင်း။ | .bold {} Class Attribute တန်ဖိုး “bold” ပိုင်ဆိုင် သော Element တိုင်းသက်ဆိုင်ပါတယ် eg - .. .p.bold {} Paragraph ၏ Class Attribute တန်ဖိုး “bold” ပိုင်ဆိုင်သော Element တိုင်းနဲ့ သက်ဆိုင်ပါတယ်။ eg - <p class="bold"> ... </p> (Element နဲ့ “.” ကပ်ရေးသောရေးထံး သည် ထို Element နဲ့ Class တွင်သာ သက်ရောက်သည်။) |

| ရွေးချယ်မှု - Selector | အဓိပါယ် | ဥပမာ |
|---------------------------|---|---|
| CHILD SELECTOR | Element အတွင်းရှိသော Child Element နှင့်တိုက်ရှိက် Applied လုပ်ခြင်း။ | li > a {} Element ၏ Child Element ဖြစ်သော တိုက်ရှိက် <a> Element များနှင့်သာသက်ဆိုင်ပါသည်။ eg - .. (ဥပမာ - အဲ့ခြား Nested Element အတွင်းကျေသော <a> Element များကို တော့ သက်ဆိုင်မည်ဟုတ် Eg - <i></i>) |
| DESCENDANT SELECTOR | Element အတွင်းရှိသော Selected Element အားလုံးကို Applied လုပ်ခြင်း။ | p a {} Element <p> အတွင်းရှိတော့ Element <a> အားလုံးအပေါ် သက်ရောက်ပါသည်။ eg - <p> .. </p> Nested Element များလည်း သက်ရောက်နိုင်ပါသည် 2nd eg - <p> ..</p> |
| ADJACENT SIBLING SELECTOR | Next Sibling Selector ပါ ဘေးကပ်ရပ် Element ကိုသာ Applied လုပ်ခြင်း။ | h1+h2 {} Element <h1> ဘေးကပ်ရပ်ရှိ <h2> Element တစ်စုံသာ သက်ဆိုင်ပါမည်။ (အဲ့ခြား <h2> Element များမပါပါ) eg - <h1> Hello </h1><h2> .. </h2> |
| GENERAL SIBLING SELECTOR | Next Sibling Selector ပါဘေးကပ်ရပ်သာမက Selected လုပ် ထားတဲ့ Element တိုင်းကို Applied လုပ်ခြင်း။ | h2~p {} Element <h2> ဘေးကပ်ရပ်ရှိ any <p> Element ကို သက်ဆိုင်ပါမည်။ eg - <h2> Hello </h2><p></p><p></p> (အဲ့ခြား <p> Element နှစ်ခု သုံးစုံမက အပေါ်သက်ရောက်မှုရှိ) |

CSS Universal Selector

ပထမအနေနဲ့ ကျွန်တော်က New Folder (ဖိုလ်ဒါအသစ်) "16_css_selector" လို့
ပေးလိုက်ပါတယ်။ နောက်ထပ် New Folder (ဖိုလ်ဒါအသစ်) "1_universal" ဆိုပြီး
တည်ဆောက်လိုက်ပါမယ် အဲဒေတွင်းမှာမှ "index.html" နဲ့ "style.css" ဆိုပြီး HTML နဲ့
CSS ဖိုင်တစ်ခုခဲ့ တည်ဆောက်လိုက်ပါမယ်။ အောက်ကပုံအတိုင်းဖြစ်သွားမှပါ။



16_css_selector (Folder) ထဲတွင်
1_universal (Folder) မူးပေါ်ထိုအထွေး
index.html & style.css ဖိုင် (ယုံကြည်)
ရှိပေါ်

VS code တွင် ပုံစံကအလိုင်း
ပေါ်နေပါမည်

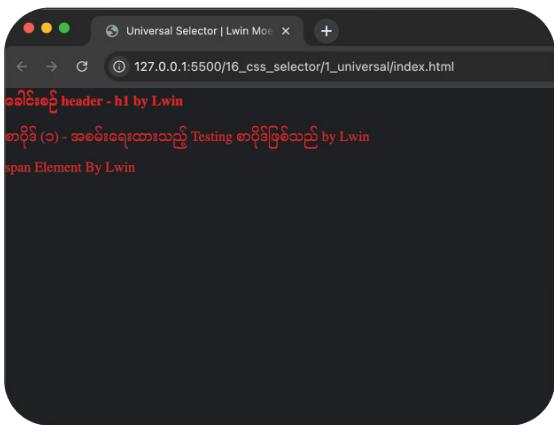
index.html ထဲတွင် အောက်က Code အတိုင်းရေးသားလိုက်ပါမည်။

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Universal Selector | Lwin Moe Paing</title>
    <link rel="stylesheet" href="style.css" type="text/css" />
</head>
<body>
    <h1>ငါးဝါး header - h1 by Lwin</h1>
    <p>တိုင် (၁) - အစမ်းရေးတားသည့် Testing စာပိုဒ်ပြုပေါ်ထဲ by Lwin</p>
    <span> span Element By Lwin </span>
</body>
</html>
```

style.css ထဲတွင် အောက်က Code အတိုင်းရေးသားပြီး Browser တွင် Run ပေးပါ။

```
* {
    color: red;
    padding: 0;
    margin: 0;
    font-size: 16px;
    background-color: #202124;
}
```

style.css ထဲတွင် အောက်က Code အတိုင်းရေးသားပြီး Browser တွင် Run ပေးပါ။



ကျွန်တော်တို့ ရေးသားလိုက်သော Element တိုင်း၏ စာသားမှာ အနီရောင်ဖြစ်သွားပြီး Element တိုင်းရဲ့ Background အရောင်မှာလည်း HexColor "#202124" ဖြစ်သွားမှာပါ။ စာဖတ်သူတို့ သတိပြုကြည့်ပါ <h1> Element က Heading ထဲမှာ အကြီးဆုံးဖြစ်ပေါ်မယ့် အာခု သုကိုယ်တိုင် font size က 16px ဖြစ်နေသောကြောင့် သူကပါ CSS Rule သက်ရောက်နေတာကို တွေ့ရမှာပါ။ padding နဲ့ margin က တော့ ကျွန်တော် နောက်အခန်းတွေမှာရှင်းပြသွားပါမယ်။

ဒီတော့ Universal Selector က Page မှာရှိတဲ့ Element အားလုံးကို Rule အရ သတ်မှတ်ချင်တဲ့အဓမ္မမှာ အသုံးပြုပါတယ်။ ရေးသားပုံကတော့ eg - * { background-color: red }

Star အနောက်မှာ “ {} ” လက်သည်းကွင်းကြားထဲ Css Rule တွေရေးသားသတ်မှတ်ပေးရတာပဲဖြစ်ပါတယ်။

Codepen ကိုအောက်က လင်ခဲ့ကနေသွားရောက်စမ်းနိုင်ပါသည်။

<https://codepen.io/lwinmoepaing/pen/mdQwMaB>

Gist Code ကိုအောက်လင့်ကနေ သွားရောက် ကူးယူနိုင်ပါတယ်

<https://gist.github.com/lwinmoepaing/b04487bde28d0a337591471360dd73ca>

<https://gist.github.com/lwinmoepaing/cca9eb16e748d84158c4a62e5e6d155f>

CSS Type Selector

Folder (ဖိုလ်ဒါ) "16_css_selector" ထဲမှာပဲ နောက်ထပ် New Folder (ဖိုလ်ဒါအသစ်) "2_type" ဆိုပြီးတည်ဆောက်လိုက်ပါပယ်။ အဲအတွင်းမှာမူ "index.html" နဲ့ "style.css" ဆိုပြီး HTML နဲ့ CSS ဖိုင်တစ်ခုဆါး တည်ဆောက်လိုက်ပါပယ်။

ကျွန်ုပ်တော်တို့ အရင်ကဗာသည်းက အသုံးပြုပြီးသားပါ example -
h1, h2, h3 { } ပုံစံနဲ့ Selector ကို Element တွေရွေးပြီးရေးသားခဲ့ပြီးသားဖြစ်ပါတယ်။
ဒုက္ခကြားငွေ့ပိုပြီး အသေးစိတ်အနေနဲ့ သိရအောင် Code တွေ ဖော်ပြထားပါတယ်။
index.html ထဲတွင် အောက်က Code အတိုင်းရေးသားလိုက်ပါမည်။

```
● ● ● 16_css_selector/2_type/index.html By Lwin Moe Paing

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Type Selector | Lwin Moe Paing</title>
    <link rel="stylesheet" href="style.css" type="text/css" />
  </head>
  <body>
    <h1>ခေါင်းစဉ် header - h1 by Lwin</h1>

    <p>ကြေးထောက တုပိုဒကို ရေးသည် !!</p>

    <h1>ဒုတိယ ခေါင်းစဉ် header - h1 by Lwin</h1>

    <span>span Element ..LMP</span>

    <h2>တတိယ ခေါင်းစဉ် header - h2 by Lwin</h2>

    <h3>ဝေါ်ထွေ ခေါင်းစဉ် header - h3 by Lwin</h3>
  </body>
</html>
```

style.css ထဲတွင် အောက်က Code အတိုင်းရေးသားပြီး Browser တွင် Run ပေးပါ။

```
● ● ● 16_css_selector/2_type/style.css

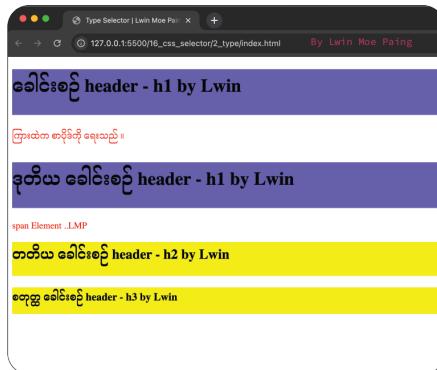
h1 {
    background-color: slateblue;
}

p, span {
    color: red;
}

h2, h3 {
    background-color: yellow;
}
```

By Lwin Moe Paing

Browser ပေါ်တွင် အောက်ကပုံအတိုင်း ထွက်ပါမည် ...



ရေးသားထားသော CSS အရ “`h1 { }`” နဲ့ Style Rule ရေးထားတဲ့အတွက် `h1` Element တိုင်းက Slate Blue Background Color ပြောင်းသွားပါတယ်။ ပြီးတော့ `p`, `span` လို့ “,” ခံရေးထားတာက ဒဲ Element () မျိုးစလုံးအတွက် သဘောဖြစ်ပါတယ်။ ထို Element () မျိုးလုံးအတွက် Text Color က အနီဖါလို့ သတ်မှတ်ထားတဲ့ အတွက် Browser မှာ ကြားထဲစာပိုဒ်ရယ် Span Element စာသားရယ်က အနီအဖြစ်ပြသွားပါတယ်။ `h2`, `h3` လည်း ထိုနည်းကဲ့သို့ပါပဲ () မျိုးလုံးအတွက် background color ကို အပါ အဖြစ် ရေးထားသောကြုံင့် Browser ရဲ့ အောက်ဆုံး header () ခုက အပါ ရောင် နောက်ခံကို ရရှိသွားတာပါ။ ဒီလောက်ဆုံး Type Css Selector ကို နားလည် လောက်ပြီလို့ ယူဆပါတယ်။

Codepen ကိုအောက်က လင်ခိုကနေသွားယောက်စမ်းနိုင်ပါသည်။

<https://codepen.io/lwinmoepaing/pen/jOQwdLj>

Gist Code ကိုအောက်လင့်ကနေ သွားရောက် ကူးယူနိုင်ပါတယ်

<https://gist.github.com/lwinmoepaing/e786a676e422c9abb6b2450e3de258ce>

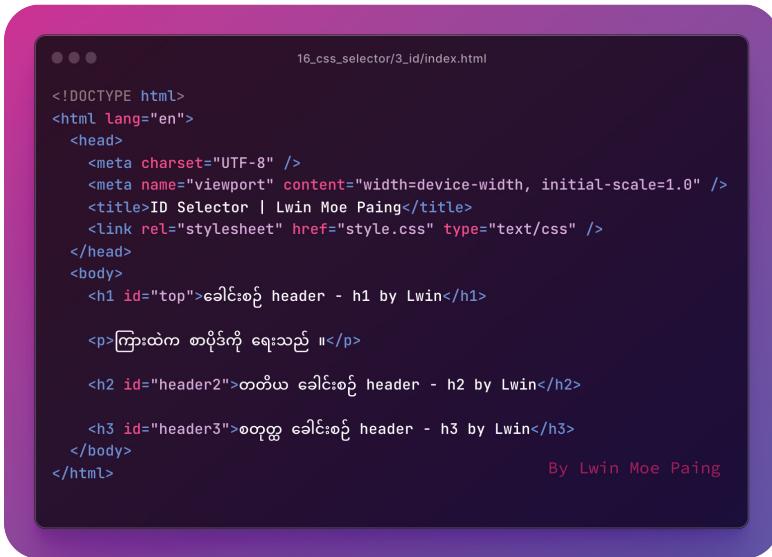
<https://gist.github.com/lwinmoepaing/397d8012742c2abbdab9c0695f718711>

CSS ID Selector

Folder (ဖိုလ်ဒါ) "16_css_selector" ထဲမှာပဲ နောက်ထပ် New Folder (ဖိုလ်ဒါအသစ်) "3_id" ဆိုပြီးတည်ဆောက်လိုက်ပါမယ်။ အဲအတွင်းမှာမူ "index.html" နဲ့ "style.css" ဆိုပြီး HTML နဲ့ CSS ဖိုင်တစ်ခုဆို တည်ဆောက်လိုက်ပါမယ်။

#id { } ပုံစံနဲ့ Selector ကို "#" sign ကိုအသုံးပြုးရေးတဲ့ ရေးသားနည်းဖြစ်ပါတယ်။ သူက Element မှာရှိတဲ့ Attribute "id" ကို ရည်ညွှန်းတာဖြစ်ပါတယ်။

index.html ထဲတွင် အောက်က Code အတိုင်းရေးသားလိုက်ပါမည်။



```

16_css_selector/3_id/index.html

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>ID Selector | Lwin Moe Paing</title>
    <link rel="stylesheet" href="style.css" type="text/css" />
  </head>
  <body>
    <h1 id="top">ခေါင်းစဉ် header - h1 by Lwin</h1>
    <p>ကြော်းထဲက စာပိုဒ်ကို ရေးသည်။</p>
    <h2 id="header2">ထာတို့ ခေါင်းစဉ် header - h2 by Lwin</h2>
    <h3 id="header3">စုတို့ ခေါင်းစဉ် header - h3 by Lwin</h3>
  </body>
</html>

```

By Lwin Moe Paing

style.css ထဲတွင် အောက်က Code အတိုင်းရေးသားပြီး Browser တွင် Run ပေးပါ။



```

16_css_selector/3_id/style.css

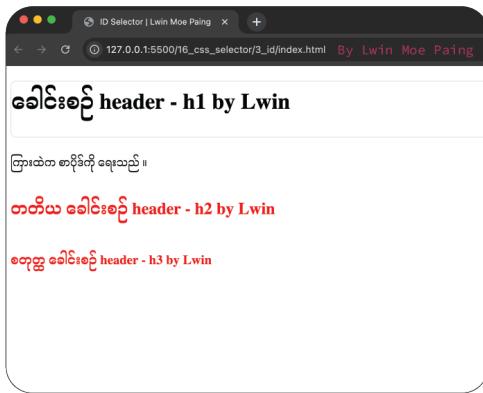
#top {
  border: 1px solid #dfdfdf;
  border-radius: 8px;
}

#header2,
#header3 {
  color: red;
}

```

By Lwin Moe Paing

Browser ပေါ်တွင် အောက်ကပုံအတိုင်း ထွက်ပါမည် ...



အရင်ဆုံး style.css ထဲကို **[ကြည့်မယ်ပါ]** '# Sign လေးကိုတွေတဲ့အပါ သူ့နောက်က Selector သည် Attribute id ရဲ့ တန်ဖိုးနဲ့ တူတဲ့ Element ကို ရွှေးချယ်ပါတယ်။ အဲတော့ #top ဆိုတာက id="top" ပိုင်ဆိုင်ထားသော Element ကို CSS rule တွေသတ်မှတ်လိုက် တာဖြစ်ပါတယ်။ HTML code ထဲမှာ <h1 id="top"> ဖြစ်တဲ့အတွက် သူ့ကို border 1px နဲ့ Solid style အရောင် #fdfdfdf Hex color ကို ပေးလိုက်တာကို မြင်ရမှာပါ။

နောက်ထပ် #header2, #header3 က “,” comma ကိုနဲ့ပိုင်းထားသောကြောင့် ထို id element နှစ်ခုလုံးကို style သတ်မှတ်ပေးခြင်းဖြစ်ပါတယ်။ “color: red” ဆို သောကြောင့် စာသားများကို အနီရောင်ပြောင်းပေးမှာဖြစ်ပါတယ်။ ကျွန်တော် တို့ HTML code ထဲမှာ သွားကြည့်လိုက်ရင် <h2 id="header2"> ဖြစ်ပြီး <h3 id="header3"> ဖြစ်သောကြောင့် ထို Element ၂ ခုလုံးအပေါ်ကို စာသား အနီရောင်က သက်ရောက်သွားမှာဖြစ်ပါတယ်။ ဒါဆိုရင် Id နဲ့ CSS မှာ Select လုပ်တဲ့ပုံကို နားလည် သွားပြီလို့ ထင်ပါတယ်။

Codepen ကိုအောက်က လင်နဲ့ကနေသွားယောက်စမ်းနိုင်ပါသည်။

<https://codepen.io/lwinmoeapaing/pen/NWEgoYV>

Gist Code ကိုအောက်လင့်ကနေ သွားရောက် ကူးယူနိုင်ပါတယ်

<https://gist.github.com/lwinmoeapaing/5b3109a56d9d69d4658aaa7dfb00054f>

<https://gist.github.com/lwinmoeapaing/981eecedb64591ec5a3cb81c946a706c>

CSS Class Selector

Class Selector က အသုံးအများဆုံး Selector ဖြစ်တဲ့အတွက် သေခါ့ ရရှိကိုပြီး လိုက်လုပ်ကြည့်ပေးပါ။ စာဖတ်သူတို့ နောက်ပိုင်း Bootstrap Css Library တို့ Tailwind Css တို့ကို အသုံးပြုတဲ့အပါ CSS class selector တွေကို များစွာ အသုံးပြုကြရမှာဖြစ်ပါတယ်။

Folder (ဖိုလ်ဒါ) "16_css_selector" ထဲမှာပဲ နောက်ထင် New Folder (ဖိုလ်ဒါအသစ်) "4_class" ဆိုပြီးတည်ဆောက်လိုက်ပါမယ်။ အဲအတွင်းမှာမူ "index.html" နဲ့ "style.css" ဆိုပြီး HTML နဲ့ CSS ဖိုင်တစ်ခုဆီ တည်ဆောက်လိုက်ပါမယ်။

index.html ထဲတွင် အောက်က Code အတိုင်းရေးသားလိုက်ပါမည်။

```
16_css_selector/4_class/index.html
By Lwin Moe Paing

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>ID Selector | Lwin Moe Paing</title>
    <link rel="stylesheet" href="style.css" type="text/css" />
  </head>
  <body>
    <div class="container border">
      <h1 class="bg-default">Title By Lwin</h1>

      <p class="underline">
        မိုးသာဝါ စာပိုဒ်အထူး Underline Text
        <span class="text-red"> By Lwin</span>
      </p>

      <span class="underline">
        Span element နဲ့ UnderLine Text
        <span class="text-red"> By Lwin</span>
      </span>
    </div>

    <div class="container">
      <p class="bold border">
        <span class="text-red">Bold</span>
        သုတေသနသာ စာပိုဒ်
        <span class="text-red">By Lwin</span>
      </p>
    </div>
  </body>
</html>
```

style.css ထဲတွင် အောက်က Code အတိုင်းရေးသားပြီး Browser တွင် Run ပေးပါ။



```

16_css_selector/4_class/style.css

.container {
    max-width: 400px;
    margin: 0 auto;
    padding: 1rem;
}

.border {
    border: 1px solid #dfdfdf;
    border-radius: 4px;
}

.bg-default {
    background-color: #dfdfdf;
}

.bold {
    font-weight: bold;
}

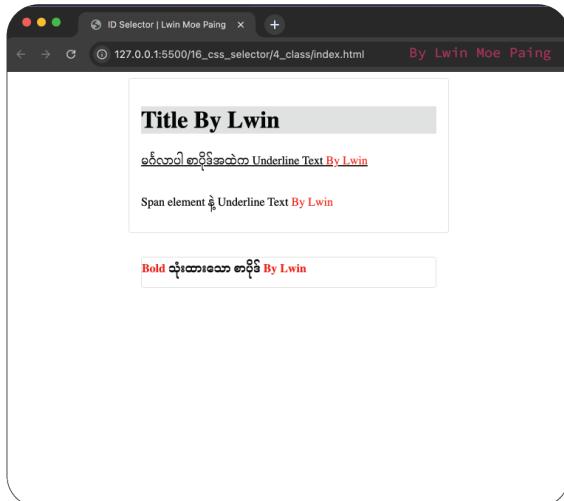
.underline {
    text-decoration: underline;
}

.text-red {
    color: red;
}

```

By Lwin Moe Paing

Browser ပေါ်တွင် အောက်ကပုံအတိုင်း ထွက်ပါမည် ...



ကျွန်ုတ်တို့ ဒီအပိုင်းက အရေးကြီးတဲ့အတွက် ဖော်ဖော်ချင်းလေ့လာကြည့်ရအောင်။ အရင်ဆုံး Code တွေကို ရေးရတာခက်နေရင် နောက်တစ်မျက်နှာမှာ Gist Code တွေ ကို ဖော်ပြပေးထားပါတယ် ကူးယူပြီး စာဖတ်ဘူတိစက်ထဲ အရင် Run ပေးပါ။

Codepen ကိုအောက်က လင်ခဲ့ကနေသွားယောက်စမ်းနိုင်ပါသည်။

<https://codepen.io/lwinmoepaing/pen/RwqqgvyJ>

Gist Code ကိုအောက်လင့်ကနေ သွားရောက် ကူးယူနိုင်ပါတယ်

<https://gist.github.com/lwinmoepaing/31797b4cc426030f1259615d0f218b26>

<https://gist.github.com/lwinmoepaing/2130dbdb56e92b39782676825bb98418>

Class Selector “ . ” (FullStop) အနောက်ရှိတဲ့ နာမည်နဲ့ တူသော Class Elements များအပေါ်သက်ရောက်ပါတယ်။ “ . ” နောက်မှာ Space မြဲးပါဘူးနော် ကပ်ရေးထာ ကို သိပြုပေးပါ။

ပထမဦးစွာ style.css ထဲက Rules တွေကို အရင်သွားလေ့လာရအောင်

- .container မှာသတ်မှတ်ထားတာက maximum width (အများဆုံးအကျယ်) ကို 400px ပဲထားမယ်လို့သတ်မှတ်ထားပါတယ်။ margin: 0 auto အလယ်ရောက်အောင် Align ချထားတာဖြစ်ပြီး padding: 1rem ကတော့ အတွင်းက content တွေကို ကပ်မသွားအောင် spacing နည်းနည်းစွားထားတာပါ။ Margin နဲ့ Padding ကို နောက်ပိုင်းအခန်းတွေမှာ အသေးစိတ်ဖော်ပြသွားပါမယ်။
- .border မှာသတ်မှတ်ထားတာက 1px solid #fdfdfd ဆိုတာက 1px အကျယ်ရှိသော Solid border style နဲ့ border လိုင်း အရောင် HexColor #fdfdfd ဖြစ်ပါတယ်လို့ သတ်မှတ်ထားခြင်းဖြစ်ပါတယ်။
- .bg-default ဆိုတဲ့ Class ကတော့ Background Color နောက်ခံအရောင်ကို HexColor #fdfdfd အရောင်ထားမယ်လို့ သတ်မှတ်ထားခြင်းဖြစ်ပါတယ်။
- .bold ကိုသုံးမယ်ဆိုရင် စာသား Font Weight က ထူးသွားမှာပါ။
- p.underline ဆိုတဲ့ အရေးအသားကို သတိထားကြည့်ပါ။
ပဲရဲ့အနောက်မှာ FullStop “.” Sign ရှိပြီး သူရဲ့အနောက်မှာ Class နာမည် underline ရေးသားထားပါတယ်။ Space မြဲးထားထားကို သတိပြုပေးပါ။ အမို့ပြုယ်က element <p> ရဲ့ class “underline” ကိုပဲ သက်ရောက်မှာဖြစ်ပါတယ်။
eg - <p class="underline"> အေား element တွေရဲ့ class="underline" အပေါ် သက်ရောက်မှုရှိမှာမဟုတ်ပါဘူး။ Style Rule အရတော့ Text Decoration က စာသားရဲ့ ဖော်ပြုမှုကို underline တားပယ်ဆိုတဲ့ပုံစံဖြင့် သတ်မှတ်ထားတာပါ။
- .text-red ကတော့ စာသားအရောင် အနီးအနှစ်အဖြစ်ပြမယ်လို့ သတ်မှတ်ထားခြင်း ဖြစ်ပါတယ်။

ဒါဆိရင် ကျွန်တော်တို့ ရေးသားထားတဲ့ HTML Code တွေဘက်ကို လေ့လာကြည့် ဖို့ အချိန်ရောက်ပါပြီ။ အရင်ဆုံး body Element အတွင်းမှာ ပထမဆုံးတွေ့ရတယ် Element ကတော့ <div> Element ဖြစ်ပါတယ်။ <div> Element ကို (J) ခါသုံးထားကြောင်းတွေ့ရမှာပါ။ အောက်က ပုံလေးကိုတစ်ချက်အရင်ကြည့်ပေးပါ ...

```
<div class="container border">
...
</div>
```

```
<div class="container">
...
</div>
```

By Lwin Moe Paing

ထဲမ <div> Element က Class နှစ်ခုကို သုံးထားပါတယ် class="container border" ဆိုပြီး မြင်တွေ့ရမှာပါ။ သတိပြုရမှာပါ Element တစ်ခုမှာ “.” နဲ့ ရေးထားတဲ့ Class-Style Rule တွေကို တစ်ခုထက်မက အသုံးပြန်ပါတယ်။ အလယ်မှာ Space လေးခြား ပြီး class တွေကို ထပ်ပေါင်းရေးသားနိုင်ပါတယ်။

<div class="container border"> ဆိုတဲ့အဓိပ္ပာယ်က ကျွန်တော်တို့ style.css အတွင်းမှာ ရှိတဲ့ .container နဲ့ .border ကို အသုံးပြုမယ်ဆိုတဲ့ သဘောတရားပါ။ ဒါကြောင့် ပထမ div element က Border နဲ့ Container Class ကိုယူသုံးထားတဲ့အတွက် အလယ်မှာ ရှိ နေပြီး ပုံပြုကြောင်း Border ပါသက်ရောက်မှုရှိနေမှာပါ။ ဒုတိယ div element ကတော့ container class ပဲ အသုံးပြုထားတာကြောင့် အလယ်ကို ပဲရောက်နေမှာဖြစ်ပါတယ်။

style.css ထဲမှာ ရေးထားတဲ့ .text-red ကို အသုံးပြုထားတဲ့ Element တိုင်းကတော့ စာသားအနီလေးတွေကြောင်းနေမှာပါ။ ဥပမာ - By Lwin တွေအားလုံးပါဝင်တဲ့ တွေကို သက်ရောက်မှုရှိပါတယ်။

ထဲမ div element အတွင်းမှာရှိတဲ့ <h1 class="bg-default"> Header element က .bg-default ကိုယူသုံးထားတဲ့အတွက် Background Color က HexColor #fdfdfc ကို ရရှိနေမှာပါ။ ဒီလောက်ဆို class အသုံးပြုနည်းကို အကြမ်းဖျည့်နားလည်သွားပြီလို ယူဆလိုက်ပါပြီ။ ထူးခြားချက်အနေနဲ့ p.underline ကို နောက်တစ်မျက်နှာမှာ လေ့လာကြည့်ရအောင်။

Space မျှေးထားသဲ ပဲရဲ့အနောက်မှာ FullStop “.” Sign ရှိပြီး သူရဲ့အနောက်မှာ Class နာမည် underline ရေးသားထားပါတယ်။ ထိုသို့ရေးသားနည်းက ပဲ element နဲ့ပဲသက်ဆိုင်သော class လို့ အဓိပ္ပာယ်ရပါတယ် အောက်က ပုံလေးကို ကြည့်ကြည့်ပါ။

```
<p class="underline">
    ...
</p>           p.underline

<span class="underline">
    ...
</span>           By Lwin Moe Paing
```

အပေါက Paragraph element ကိုပဲ အောက်မျဉ်းကြောင်း Underline Style က သက်ရောက်မှုရှိမှာပါ။ အောက်က span element နဲ့တော့ သက်ဆိုင်မှုရှိမှာမဟုတ်ပါဘူး။ ဒီလေးကိုရင် ကွဲနိုင်တယ်တို့ Class Selector အသုံးပြန်ည်းကို လေ့လာလို့ ပြီး သွားပြီပဲဖြစ်ပါတယ်။

CSS Child Selector

Element အတွင်းရှိသော Child Element နှင့်တိုက်ရှိက် Style များကို Applied လုပ်သော ပုံစံပါ။

Folder (ဖိုလ်ဒါ) “16_css_selector” ထဲမှာပဲ နောက်ထပ် New Folder (ဖိုလ်ဒါအသစ်) “5_child” ဆိုပြီးတည်ဆောက်လိုက်ပါမယ်။ အဲအတွင်းမှာမ “index.html” နဲ့ “style.css” ဆိုပြီး HTML နဲ့ CSS ဖိုင်တစ်ခုဆီ တည်ဆောက်လိုက်ပါမယ်။

သူရဲ့ရေးသားနည်းမှာ သတိပြုရမှာက “>” Greater than Sign ကိုအသုံးပြုတာပါပဲ Child Selector ဆိုတဲ့အတိုင်းပဲ သူရဲ့ အတွင်းမှာရှိတဲ့ တိုက်ရှိက် ကလေး (သားသမီး) နဲ့ ပဲတိုက်ဆိုင်မယ်ဆိုတဲ့ သဘောပါ။ နောက်တစ်မျက်နှာမှာ Code တွေတို့ အရင်ကူးယူပြီး စာဖတ်သူတို့ စက်ထဲတွင် အရင် Run ပေးပါ။

Codepen ကိုအောက်က လင်ခဲ့ကနေသွားပောက်စမ်းနိုင်ပါသည်။

<https://codepen.io/lwinmoeapaing/pen/YQZbqV>

Gist Code ကိုအောက်လင့်ကနေ သွားရောက် ကူးယူနိုင်ပါတယ်

<https://gist.github.com/lwinmoeapaing/29107b24898c0862774e700827f128cb>

<https://gist.github.com/lwinmoeapaing/8dcee0d3f0cc4b32292c7c0d34c893ee>

index.html ထဲတွင် အောက်က Code အတိုင်းရေးသားလိုက်ပါမည်။

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Child Selector | Lwin Moe Paing</title>
    <link rel="stylesheet" href="style.css" type="text/css" />
</head>
<body>
    <h2>Lwin's Cafe</h2>

    <ul>
        <li>Coffee</li>
        <li>Milk</li>
    </ul>

    <section>
        <p>Paragraph 1</p>
        <p>Paragraph 2</p>

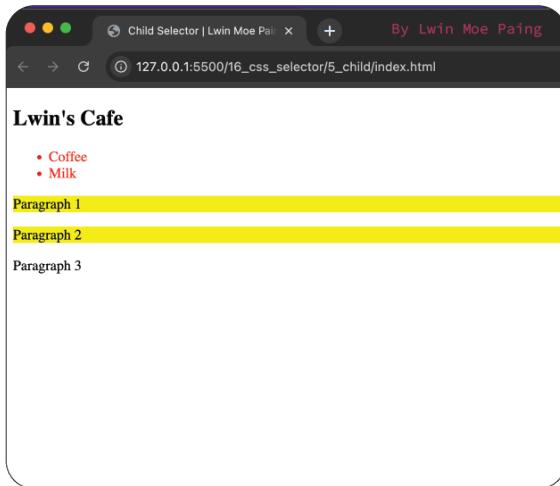
        <div>
            <p>Paragraph 3</p>
        </div>
    </section>
</body>
</html>
```

style.css ထဲတွင် အောက်က Code အတိုင်းရေးသားပြီး Browser တွင် Run ပေးပါ။

```
ul > li {
    color: red;
}

section > p {
    background: yellow;
}
```

Browser ပေါ်တွင် အောက်က ပုံအတိုင်း မြင်ရပါမည်။

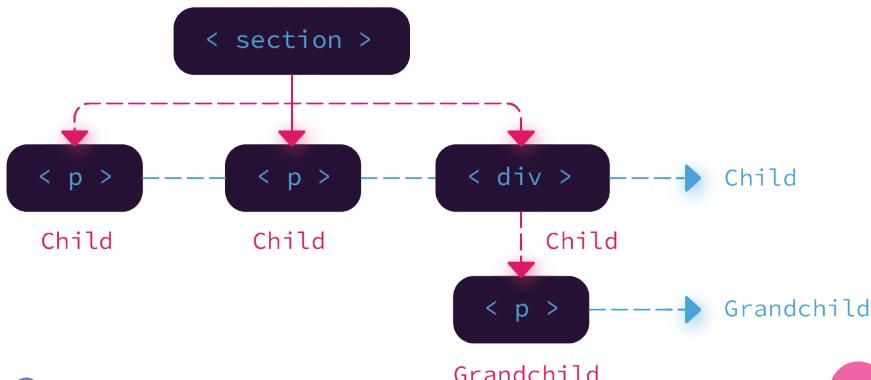


အရင်ဆုံး style.css ထဲကို ကြည့်မယ်ပါ '>' လေးကို မြင်ရမှာဖြစ်ပါတယ် သူရဲ့ အနီးဆုံး ဆက်ခံမယ့် Child ဖြစ်တဲ့ (သားသမီး) Element ကိုပဲ သက်ရောက်မှာဖြစ်ပါတယ်။

ထုတေသနများတွင် `` အတွင်း ရှိ `` List items Element များကို color:red ဆိုပြီး ပေးထားတာကိုမြင်ရမှာပါ။ အဲဒေသတိုင်းပဲ Browser မှာ Run ပေးလိုက်ပါတယ်။

ဒုတိယ Paragraph ပေးထားတဲ့ Style Rule အရ `section > p` ဆိုပြီး ရေးထားတာကို တွေ့ရမှာပါ။ Browser မှာ Run ထားတာကို Section ထဲမှာ `<p>` Element (၃) ခုရှိရဲ့ သားနဲ့ သာလို့ (၂) ခုကိုပဲ သက်ရောက်သွားတာလ ?
အောက်က ပုံလေးကို တစ်ချက်ကြည့်ပေးပါ။

By Lwin Moe Paing



<section> Element ရဲ့ ပထားတွေရတဲ့ <p> Element (J) ခုက တိုက်ရှိက် သူ့ရဲ့ Child (သားသမီး element) ဖြစ်ပါတယ်။ တတိယောက်တွေ့ရတယ် <p> element ကတော့ <div> element ရဲ့ အထဲရောက်နေတာပါ။ ဆိုလိုချင်တာက <div> element ကတော့ <section> element ရဲ့ child (သားသမီး) element ပါ။ သူ့ရဲ့ ကလေးဖြစ်တယ့် <p> element က တော့ <div> element ရဲ့ child element ဖြစ်သွားပါတယ်။ အဲ တော့ <section> element နဲ့ (ဂ) ခုမောက် <p> element တို့ ဆက်ခံပဲက သူ့ရဲ့ grandchild element (ငြေား element) ဖြစ်သွားတယ့်အတွက် Style Rule က သက်ရောက်မှုပရှိတော့ပါဘူး။ အခါလို ပုံကိုသွားကြည်ပြီး လေ့လာမယ်ဆိုရင် သဘောပေါက်သွားလိမ့်မယ်လို့ မွှေ့ကြလင့်ပါတယ်။

CSS Descendant Selector

Element အတွင်းရှိသော Selected Element များကို Applied လုပ်သော ပုံစံပါ။ Child Selector လိုမဟုတ်ဘူး အတွင်းမှာရှိတဲ့ Element များကိုလည်း သက်ရောက်မှုရှိပါတယ်

Folder (ဖိုလ်ဒါ) “16_css_selector” ထဲမှာပဲ နောက်ထပ် New Folder (ဖိုလ်ဒါအသစ်) “6_descendant” ဆိုပြီးတည်ဆောက်လိုက်ပါမယ်။ အထဲတွင် “index.html” နဲ့ “style.css” ဆိုပြီး HTML နဲ့ CSS ဖိုင်တစ်ခုဆီ တည်ဆောက်လိုက်ပါမယ်။

Codepen ကိုအောက်က လင်ခဲ့ကနေသွားယောက်စမ်းနိုင်ပါသည်။

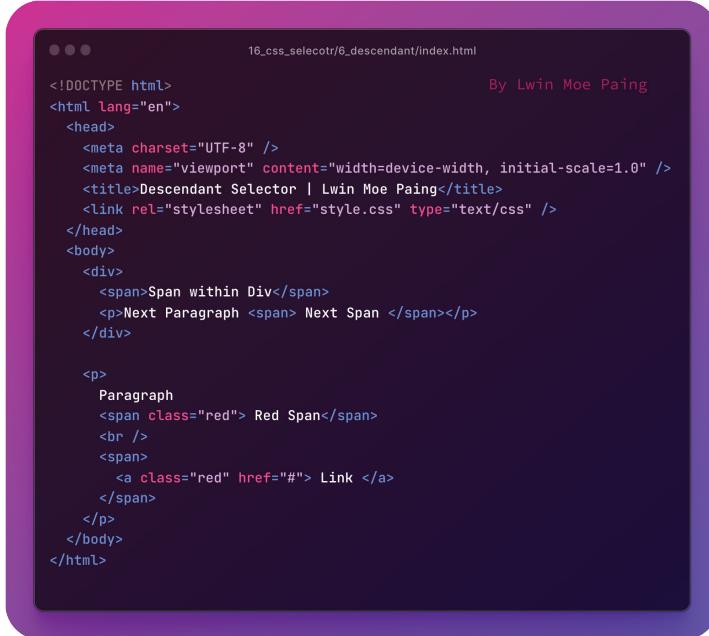
<https://codepen.io/lwinmoepaing/pen/abQweNM>

Gist Code ကိုအောက်လင့်ကနေ သွားရောက် ကူးယူနိုင်ပါတယ်

<https://gist.github.com/lwinmoepaing/8f1e56379c60e1f616245f0a6668abba>

<https://gist.github.com/lwinmoepaing/f448429b3b3b192a6163e55e50894cdf>

index.html ထဲတွင် အောက်က Code အတိုင်းရေးသားလိုက်ပါမည်။



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Descendant Selector | Lwin Moe Paing</title>
    <link rel="stylesheet" href="style.css" type="text/css" />
</head>
<body>
    <div>
        <span>Span within Div</span>
        <p>Next Paragraph <span> Next Span </span></p>
    </div>

    <p>
        Paragraph
        <span class="red"> Red Span</span>
        <br />
        <span>
            <a class="red" href="#"> Link </a>
        </span>
    </p>
</body>
</html>
```

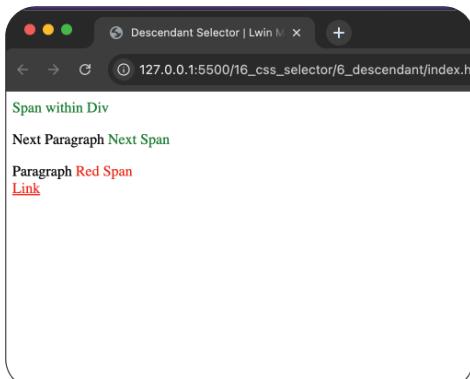
style.css ထဲတွင် အောက်က Code အတိုင်းရေးသားပြီး Browser တွင် Run ပေးပါ။



```
div span {
    color: green;
}

p .red {
    color: red;
}
```

Browser ပေါ်တွင် အောက်က ပုံအတိုင်း မြင်ရပါမည်။



Style ထဲမှာ div span { } ပုံစံနဲ့ ရေးထားတဲ့အတွက် ကြောင့် div ထဲမှာရှိတယ့် မည် သည့် element မဆို သက်ဆိုင်ပါတယ်။ အဲဒေသကို Element သူး ထားတဲ့ စာသား Content တိုင်းက အစိမ်းရောင် Style Rule ကို ရရှိမှာပါ။

ဒုတိယ Rule က p.red { } ပုံစံနဲ့ ရေးသားထားခြင်းဖြစ်ပါတယ်။ ဆိုလိုချင်တာက <p> element အတွင်း မည်သည့် class attribute "red" ရှိသော element မဆို သက်ဆိုင်ကြောင့် ကြိုးငြာထားတာဖြစ်ပါတယ်။

မှတ်ချက်။ p.red (full stop) ကြားထဲ ကပ်ရေးမိပါက p element နှင့်သာဆိုင်သော red class <p class="red" /> ဖြစ်သွားပါလိမ့်မယ်။ ဒါကြောင့် အလယ်တွင် Space ခြားနေသည် ကို သတိပြုပေးစေလိုပါတယ်။

ဒါဆိုရင် Descendant Selector အကြောင်းလေ့လာပြီးသွားပါပြီ။

CSS Adjacent Sibling Selector

Next Sibling Selector ပုံစံပါ ဘေးကပ်ရပ်ရှိသော Element ကို Select လုပ်ပြီး Style Applied ပြုလုပ်ခြင်းဖြစ်ပါတယ်။ "+" Sign လေးနဲ့ ရေးပေးရမှာဖြစ်ပါတယ်။

Folder (ဖိုလ်ဒါ) "16_css_selector" ထဲမှာပဲ နောက်ထပ် New Folder (ဖိုလ်ဒါအသစ်) "7_adjacent_sibling" ဆိုပြီးတည်ဆောက်လိုက်ပါမယ်။ အဲဒေသဗိုင်း "index.html" နဲ့ "style.css" ဆိုပြီး HTML နဲ့ CSS ဖိုင်တစ်ခုဆီ တည်ဆောက်လိုက်ပါမယ်။

Codepen ကိုအောက်က လင်ခဲ့ကနေသွားယောက်စမ်းနိုင်ပါသည်။

<https://codepen.io/lwinmoepaing/pen/RwqZbqB>

Gist Code ကိုအောက်လင့်ကနေ သွားရောက် ကူးယူနိုင်ပါတယ်

<https://gist.github.com/lwinmoepaing/b6857fbcc3502ca4e8bfd3965cf39ea6>

<https://gist.github.com/lwinmoepaing/e702118e983db3f6c3f7bfd63130587f>

index.html ထဲတွင် အောက်က Code အတိုင်းရေးသားလိုက်ပါမည်။

```
16_css_selector/7_adjacent_sibling/index.html By Lwin Moe Paing

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<title>Adjacent Sibling | Lwin Moe Paing</title>
<link rel="stylesheet" href="style.css" />
</head>
<body>
<div>
<h1>H1 Header by Lwin</h1>
<h2>H2 Header by Lwin</h2>
</div>

<div>
<h2>H2 Element by Lwin</h2>
<p>P Tag by Lwin</p>
<p>2nd tag by p</p>
<h2>H2 Element by Lwin</h2>
<p>P Tag by Lwin</p>
<p>2nd tag by p</p>
</div>
</body>
</html>
```

style.css ထဲတွင် အောက်က Code အတိုင်းရေးသားပြီး Browser စွင် Run ပေးပါ။

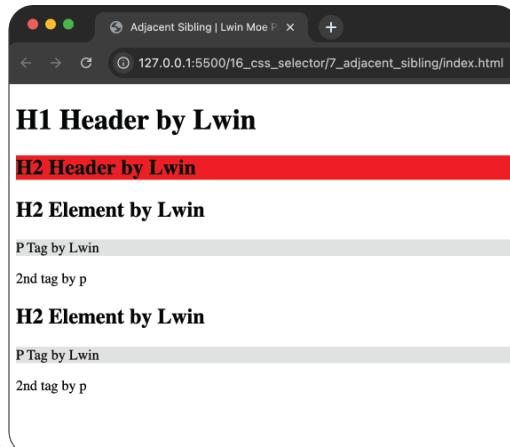
```
16_css_selector/7_adjacent_sibling/style.css

h1 + h2 {
    background: red;
}

h2 + p {
    background: #fdfdfdf;
}
```

By Lwin Moe Paing

Browser ပေါ်တွင် အောက်က ပုံအတိုင်းမြင်ရပါမည်။



Style ရေးရာတွင် “+” Sign နဲ့ ရေးရမှာဖြစ်တာကို သတိပြုပေးပါ ။

ထထမ Style က h1 + h2 ဆိုပြီးရေးထားတာကိုမြင်တွေ့ရမှာပါ ဆိုလိုချင်တာက <h1> element ဘေး (သို့) အောက်တွင် ကပ်ရပ်ရှိသော <h2> element အပေါ် rule က သက်ရောက်မှာဖြစ်ပါတယ်။ ဒါကြောင့် H2 Header By Lwin ဆိုတဲ့ <h2> element ရဲ့ Background အရောင်က အနီရောင်ဖြစ်နေတာကို တွေ့ရမှာပါ ။

ခုတိယ က h2 + p ဆိုပြီးရေးထားခြင်းဖြစ်ပါတယ် । <h2> element ရဲ့ ဘေး (သို့) အောက်တွင် ကပ်ရပ်ရှိသော <p> element ကို သက်ရောက်သွားမှာဖြစ်ပါတယ်။ သူက အောက်မှာရှိတဲ့ <div> Element ကို တစ်ချက်ကြည့်ပေးပါ ။ အတွင်းမှာရှိတဲ့ <h2> element ရဲ့ အောက်က <p> element တစ်ကိုအပေါ် သက်ရောက်နေတာ ကိုတွေ့ရမှာပါ ။ 2nd tag by p ဆိုသော <p> element ခုတိယ တစ်ခုအပေါ်ကိုတော့ အရောင်မပြောင်းလဲခြင်း မရှိပါဘူး ။

ဒါဆိုရင် ကျွန်ုတ်တို့ Adjacent Sibling Selector ကိုလေ့လိုပြီးပြီး ဖြစ်ပါတယ် ။

CSS General Sibling Selector

Next Sibling Selector ပါဘေးကပ်ရပ်သာမက Selected လုပ်ထားတဲ့ Element တိုင်း ကို Applied လုပ်ပေးတဲ့ ပုံစံဖြစ်ပါတယ် ။ သူကတော့ “~” Sign ကို အသုံးပြုပါတယ် ။

Folder (ဖိုလ်ဒါ) “16_css_selector” ထဲမှာပဲ နောက်ထပ် New Folder (ဖိုလ်ဒါအသစ်) “8_general_sibling” ဆိုပြီးတည်ဆောက်လိုက်ပါမယ် ။ အဲဒေတွင်း “index.html” နဲ့ “style.css” ဆိုပြီး HTML နဲ့ CSS ဖိုင်တစ်ခုဆီ တည်ဆောက်လိုက်ပါမယ် ။

Codepen ကိုအောက်က လင်ခဲ့ကနေသွားယောက်စမ်းနိုင်ပါသည် ။

<https://codepen.io/lwinmoeapaing/pen/dyQzomZ>

Gist Code ကိုအောက်လင့်ကနေ သွားရောက် ကူးယူနိုင်ပါတယ်

<https://gist.github.com/lwinmoeapaing/6017487897284a01f2d0352ceb19b224>

<https://gist.github.com/lwinmoeapaing/3dbd4415ddb74cd33a0f4add3c706129>

index.html ထဲတွင် အောက်က Code အတိုင်းရေးသားလိုက်ပါမည်။

```
16_css_selecotr/8_general_sibling/index.html By Lwin Moe Paing

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>General Sibling | Lwin Moe Paing</title>
    <link rel="stylesheet" href="style.css" type="text/css"/>
</head>
<body>
    <div>
        <h1>H1 Header by Lwin</h1>
        <h2>H2 Header by Lwin</h2>
        <h2>H2 Header by Lwin</h2>
    </div>

    <div>
        <h2>H2 Element by Lwin</h2>
        <p>1st P Tag by Lwin</p>
        <p>2nd tag with p</p>
        <p>3rd tag with p</p>
        <p>4rd tag with p</p>
    </div>
</body>
</html>
```

style.css ထဲတွင် အောက်က Code အတိုင်းရေးသားပြီး Browser စွင် Run ပေးပါ။

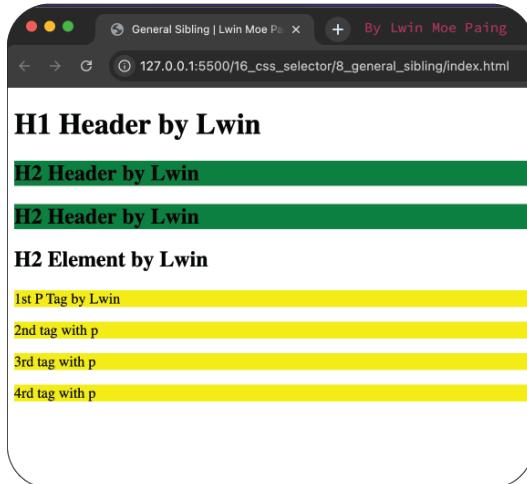
```
16_css_selecotr/8_general_sibling/style.css

h1 ~ h2 {
    background: green;
}

h2 ~ p {
    background: yellow;
}
```

By Lwin Moe Paing

Browser ပေါ်တွင် အောက်က ပုံအတိုင်းမြင်ရပါမည်။



Style ရေးရာတွင် “~” Sign နဲ့ ရေးရမှာဖြစ်တာကို သတိပြုပေးပါ။

ထထမ Style က h1 ~ h2 ဆိုပြီးရေးထားတာကိုမြင်တွေ့ရမှာပါ ဆိုလိုချင်တာက <h1> element ဘေး (သို့) အောက်တွင်ရှိသော <h2> element တိုင်းအပေါ် rule က သက်ရောက်မှာဖြစ်ပါတယ်။ ဒါကြောင့် H2 Header By Lwin ဆိုတဲ့ <h2> element (၂) ခုလုံး ရဲ့။

Background အရောင်က အစိမ်းရောင်ဖြစ်နေတာကို တွေ့ရမှာပါ။

ဒုတိယ က h2 ~ p ဆိုပြီးရေးထားခြင်းဖြစ်ပါတယ်။ <h2> element ရဲ့ ဘေး (သို့) အောက်တွင်ရှိသော <p> element တိုင်းအပေါ်ကို သက်ရောက်သွားမှာဖြစ်ပါတယ်။ ဒါကြောင့် <p> Element (၄) ခုလုံးကို Background အဝါရောင်က သက်ရောက်သွားမှာပါ။

ဒါဆိုရင် ကျွန်ုတ်တို့ General Sibling Selector ကိုလေ့လိုပြီးပြီး ဖြစ်ပါတယ်။

Congratulation ပါပျေ အခုခု တာဖတ်သူတို့က CSS ရဲ့ Selector အခန်းက ပြီးသွားပြီ ဖြစ်ပါတယ်။ တော်တော်များများ မမှတ်မိခဲ့ရင် CSS Selector လေားကိုသွားပြီး အနှစ်ချုပ်သွားဖတ်လေ့လာနိုင်ပါတယ်ဗျျှား။

နောက်တစ်မျက်နှာမှာတော့ ကျွန်ုတ်တို့ CSS Selector ကို Practical အနေနဲ့ လက်တွေ့လုပ်ဆောင်ရမှာပါ ...

From Theory to Practice: CSS Selector

အခုစိုရင် Practice လုပ်ဖို့အချိန်ရောက်လာပြန်ပါပီ CSS Selector နှုပါတ်သတိပြီး စာတော်သူတို့အနေနဲ့ ဘယ်လောက်ထိ နားလည်သွားလဲဆိုတာ စမ်းသပ်ကြရမှာဖြစ်ပါ တယ်။ ပထမအနေနဲ့ ကျွန်ုတ်က New Folder (ဖိုင်ဒါအသစ်) "17_css_practice" လို့ ပေးလိုက်ပါတယ်။ အထဲမှာ "index.html" ဆိုပြီး HTML file ရယ် နောက်ထပ် CSS File (ဖိုင်) ကို "style.css" ဆိုပြီး တည်ဆောက်လိုက်ပါမယ်။

index.html အတွက် အောက်တွင်ပေးထားသော Gist Code ကို ကူးယူလိုက်ပါ။

<https://gist.github.com/lwinmoeapaing/c3722528a7352caba4f5f35dd0ac5866>

```
17_css_practice/index.html By Lwin Moe Paing

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>CSS Practice 1 | Lwin Moe Paing</title>
    <link rel="stylesheet" href="style.css" type="text/css" />
</head>
<body>
    <div class="container">
        <h1 class="title">This is H1 Title</h1>
        <p>P tag 1 with the text <span class="text-red">Red</span></p>
        <p id="para2">P tag 2 with Green Color</p>
        <p>P tag 3 with the text <span class="text-red">Red</span></p>
    </div>

    <div class="container">
        <h2>This is H2 Title</h2>
        <p>
            P tag 1 with the text <span class="text-red">Red</span> & LightGreen
            Background Color
        </p>
        <p>P tag 2</p>
        <p class="p-bold text-purple">
            P tag 3 with FontWeight Bold & purple color
        </p>
    </div>

    <div class="container">
        <h3>This is H3 Title</h3>
        <ul>
            <li><a href="#">Milk with Bg yellow</a></li>
            <li><a href="#">Coffee with Bg yellow</a></li>
            <li>Yogurt</li>
        </ul>
    </div>

    <div class="container">
        <div>
            <i> This is i Element within Div.</i>
        </div>
        <i> Outside i Element </i>
        <ol>
            <li><i> Some of i within ol and li .</i></li>
        </ol>
    </div>
</body>
</html>
```

style.css အတွက် အောက်တွင်ပေးထားသော Gist Code ကို ကူးယူလိုက်ပါ။

<https://gist.github.com/lwinmoepaing/61fb2821740e23d4b91d92f2e18c4da3>

```

17_css_practice/style.css
By Lwin Moe Paing

* {
    box-sizing: border-box;
}

h1,
h2,
h3 {
    margin: 0.7rem 0;
}

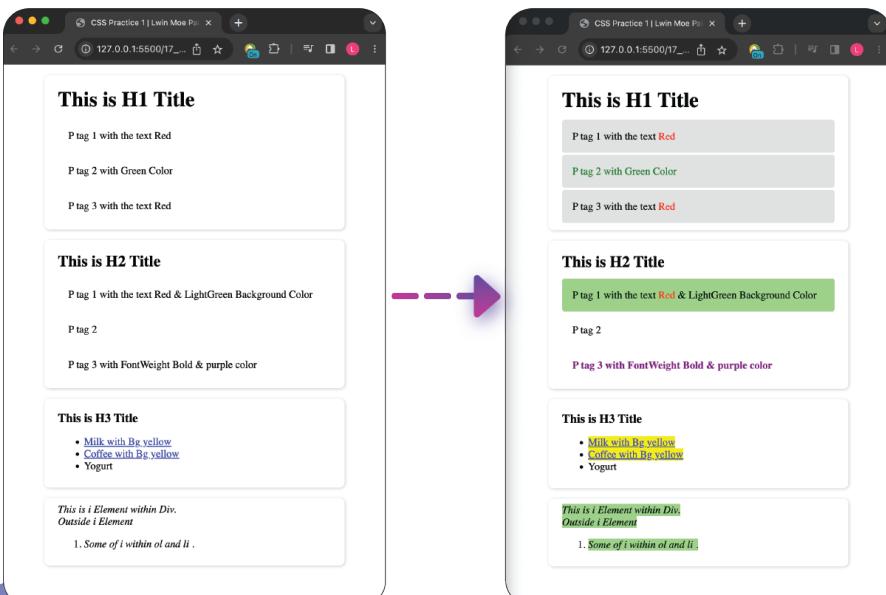
.container {
    max-width: 460px;
    margin: 1rem auto;
    border-radius: 8px;
    padding: 0.5rem 1.3rem;
    box-shadow: 1px 1px 6px #c6c6c6;
}

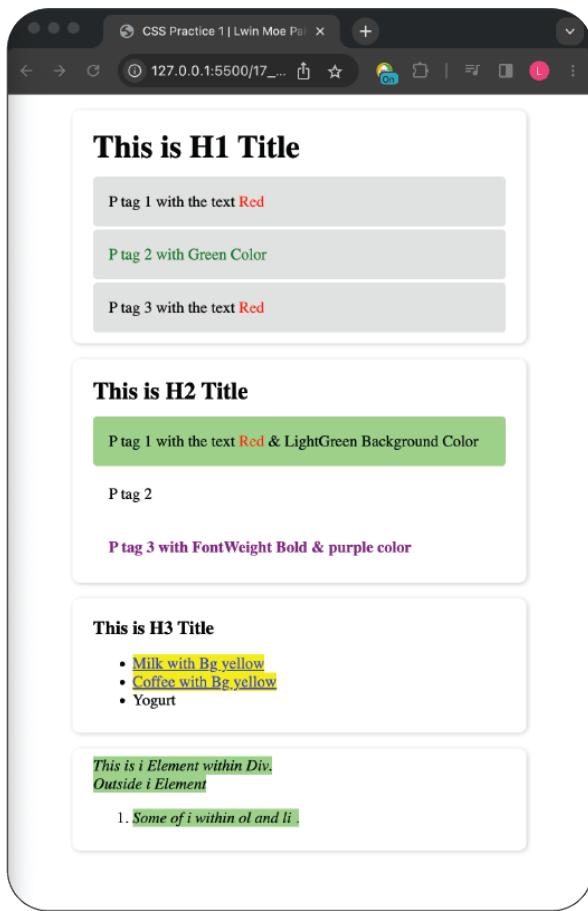
.container p {
    padding: 1rem;
    border-radius: 4px;
    margin: 0.2rem 0;
}

/* ဒေသအက်မှာ အခြေခံစွမ်းပေးပါ */

```

Browser ပေါ်တွင် Run လိုက်ပါက အောက်က ပုံ (၂) ပုံတွင် ဘယ်ဘက်က ပုံတိုင်းပေါ် နေပါလိမ့်မယ်။ style.css ထဲတွင် /* ဒေသအက်မှာ အဖြေထွေစရေးပေးပါ */ ဆိုသော Code ရဲ့ အောက်မှာ CSS style rule ထွေကို စာဖတ်သုတေသန ရေးပေးရမှာဖြစ်ပါတယ်။ အောက်က ပုံ (၂) အတိုင်းထွက်အောင် ရေးပေးရမှာပါ။





အခုတော့ အဖော်မတိုက်ခင် ကြိမ်းသေ စာဖတ်သူတို့ကိုယ်တိုင် CSS rules မပူးရေးသား နိုင်လိမ့်မယ်လို့ ယုံကြည်ပါတယ်။

ကိုယ်ပိုင်ပုံစံဖြင့် Result အရ ကောင်းမွန်စွာထွက်နေဖြီး ဖြစ်သော်လည်း ကျွန်ုတ်ပေါးထားသော Code နဲ့တူညီမှ မှန်တယ်လို့မဆိုလိုပါ။

Code ကို အောက်က Gist မှာသွားရောက် တိုက်ကြည့်ပြီးလေ့လာနိုင်ပါတယ်ၢုံး။

Gist Code for style.css

<https://gist.github.com/lwinmoeapaing/f89fc272749b2e4f9ed3d585e0046d2a>

Codepen ကို အောက်မှာတိုက်ရိုက် စမ်းနိုင်ပါတယ်။

<https://codepen.io/lwinmoeapaing/pen/poQrdQP>

CSS နှင့် Colors

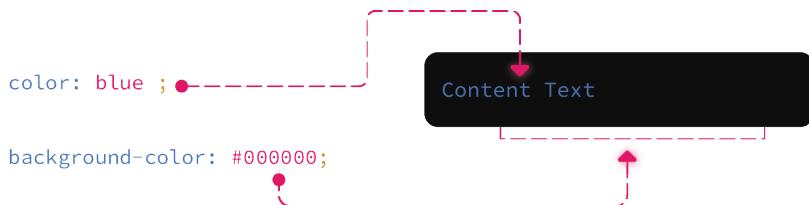
Color အရောင်တွေဆိုတာ HTML Page ကိုအသက်ဝင်သွားအောင်လုပ်မယ့်အရာဆိုလည်းများပါဘူး။ ကျွန်ုတ်တို့ ဒီအပိုင်းမှာ အရောင်တွေအကြောင်း လေ့လာကြည့်ရအောင်။

Colors တန်ဖိုးများ

CSS မှာ အရောင် (color) တန်ဖိုးတွေကို အရင် အခန်းတွေမှာလည်း သတ်မှတ်ခဲ့က သည်ကို သတိပြခဲ့ပြီးဖြစ်ပါတယ်။ အခု အသေးစိတ်လေ့လာကြည့်ရရင်

- RGB (red, green, blue) တန်ဖိုးဖြင့် သတ်မှတ်ခြင်း
- Hexadecimal (hex) code တန်ဖိုးဖြင့် သတ်မှတ်ခြင်း
- Color Names အရောင် နာမည် ဖြင့် သတ်မှတ်ခြင်း
- HSL (hue, situration, lightness) တန်ဖိုးဖြင့် သတ်မှတ်ခြင်း

အရောင်သတ်မှတ်ခြင်း (၄) မျိုးကိုတွေ့ရမှာပါ။ ဒီထဲမှာ HSL method ကတော့ အသုံးနည်းပြီး ကျွန်ုတ်တဲ့ methods (၃) မျိုးကတော့ အရမ်းကို အသုံးများပါတယ်။



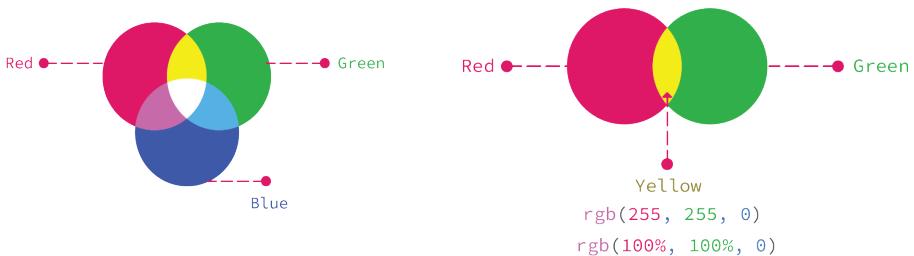
များသောအားဖြင့် တွေ့ရအများဆုံး အရောင်ရွေးချယ်မှန်စုရှိပါတယ်။ Background Color နဲ့ Foreground Color အတွက်ပါ။

Background Color ဆိုရင် "background-color" ဆိုတဲ့ Property ကိုသုံးတာ မြင်ခဲ့ပြီး သားဖြစ်ပါပြီ။ သူက Select လုပ်လိုက်တဲ့ element ရဲ့ နောက်ခံ အရောင်ကို ရွေးချယ်လိုက်တာဖြစ်ပါတယ်။

Foreground Color အတွက်ဆိုရင် "color" ကိုအသုံးပြုပြီး element အတွင်းမှာရှိတယ့် Text (သာသား) ရဲ့အရောင်တွေကိုစိမ်ပေးတာပါ။

RGB Color တန်ဖိုးအသုံးပြုခြင်း

RGB (red, green, blue) တန်ဖိုးဖြင့် သတ်မှတ်ခြင်း ကို မလေ့လာခင် Pixel ဆိတာကို အရင်သိအောင်လုပ်ကြည့်မယ်။ Pixel က Phone ထွေ Computer ထဲမှာ ရှိတဲ့ အစက် လေးတစ်စက် (သို့) လေးထောင့်ကွက်လေးတစ်ခုဖြစ်နိုင်ပါတယ်။ သူ့အတဲ့မှာ အနီ (red)၊ အစိမ်းရောင် (green)၊ အပြာရောင် (blue) ဆိတဲ့ အရောင် (၃) မျိုးကို Mixed ရောနောပြီး အမျိုးမျိုးသော အခြားအရောင်တွေကိုပါ Screen မှာဖော်ပြပေးပါတယ်။ အောက်ကပ်လေးကို တစ်ချက်ကြည့်ပေးပါ။



ကျွန်တော်တို့ ပြောခဲ့တဲ့အလိုင်း RGB method ကိုသုံးပြီးတော့ CSS မှာ အရောင်တွေ ရောစပ်အသုံးပြုနိုင်ပါတယ်။ ဘယ်ဘက် ပုံမှာဆိုရင် အရောင် (၃) ရောင်လုံးထပ်လိုက် တယ့်အပါ အဖြူရောင်ကိုရွားမှာပါ။ ညာဘက်က ပုံကတော့ အဝါရောင် (yellow color) ကိုလိုချင်တဲ့အပါ အနီ အရောင်နဲ့ အစိမ်း အရောင်ကို ဆက်စပ်ရောနောပြီးတော့ ကျွန်တော့တို့ ရယူနိုင်ပါတယ်။

ဥပမာ - `rgb(100%, 100%, 0)`

ဆိုလိုချင်တာက အနီရောင် 100% နဲ့ အစိမ်းရောင် 100% ကိုအသုံးပြုလိုက်ခြင်းဖြစ်ပါတယ်။ ရေးသားနည်းကို အောက်ကပ်လေးနဲ့ တွဲဖက် လေ့လာနိုင်ပါတယ်ဗျား။

By Lwin Moe Paing
`rgb(255, 255, 255);`

 Red Green Blue

ဆိုတော့ တန်ဖိုးအနေနဲ့ က 0 ~ 255 အထိ တန်ဖိုးတွေ အတိုးအလျော့လုပ်ပြီး ကျွန်တော်တို့က ပြောင်းလဲသတ်မှတ်နိုင်တယ် (သို့မဟုတ်) percentage ပုံစံအရေး အသားနဲ့လည်း ရေးသားနိုင်ပါတယ်။

Hexadecimal Color တန်ဖိုးအသုံးပြုခြင်း

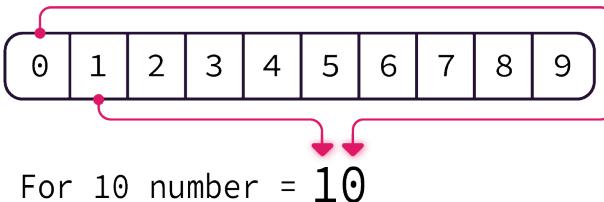
"#" sign ကိုအသုံးပြုပြီး အနောက်မှာ ကာလာကုတ်တန်ဖိုး (၆) လုံးကို ရေးသားပုံနဲ့ အသုံးပြုပါတယ်။ သူလည်းပဲ Red Green Blue ကိုအခြေခံပြီး တန်ဖိုးထည့်ခြင်းပါပဲ။ သို့ပေါ်မယ် ရှိုးရှိုး ၁ ကနေ ၁၀ အထိ တန်ဖိုးမဟုတ်ပဲ Hexadecimal တန်ဖိုးကိုအသုံးပြု ထားခြင်းပါ။

အရင်ဆုံး ကျွန်တော်တို့ သုံးနောက်ဖြစ်တယ့် Decimal Number System ကိုအောက်က ပုံလေးက ကြည့်လိုက်ပါ။ သချာဂါဏ်း အနေနဲ့ (၀) သူညှာ မှ (၉) ကိုဂါဏ်း အတိရှိပါတယ်။

Decimal Number System

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|

Base 10 တန်ဖိုးအနေနဲ့ဆို ကျွန်တော်တို့က ဂကန်းတန်ဖိုး (၉) အထိက အများဆုံးပါ။ တန်ဖိုး (၉) ကျော်သွားရင် ကျွန်တော်တို့က အရေးကို တန်ဖိုး (၁) တိုးသွားပါတယ် သို့ပေါ်မယ် အခါ ဂကန်းစဉ် ထဲက တန်ဖိုးတွေကို ပြန်သုံးပြီး ဖော်ပြပါတယ်။



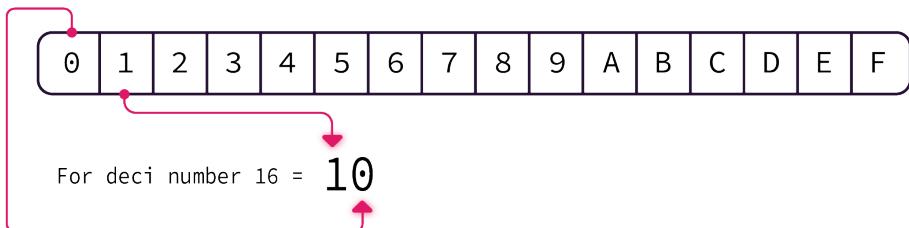
အခါ Hexadecimal Number System ကိုလေ့လာကြည့်ရအောင် သူက Base16 ဖြစ်ပါတယ်။ အခုအောက်က ပုံလေးကို ကြည့်ပေးပါ ...

Hexadecimal Number System

| | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
| 10 | 11 | 12 | 13 | 14 | 15 | | | | | | | | | | |

သူမှာက တန်ဖိုး (F) ကျွန်တော်တို့ Decimal အနေနဲ့ဆို 15 အထိ ရှိပါတယ်။ အကယ် F ထက် တန်ဖိုး ကျော်သွားတဲ့အပါ ကျွန်တော်တို့ အရေးကို တန်ဖိုး (၁) ခုထပ်တိုးသွားပါတယ်။ နောက်တစ်ယျက်နှာမှာ ထပ်လေ့လာကြည့်ရအောင် ... ။

F က ကျွန်တော်တို့ နံပါတ် System အရ (၁၅) နဲ့ ပြည့်ပါတယ်။ ဒါကြောင့် တစ်ခု ထပ်တိုးလိုက်တဲ့ အခါ ကျွန်တော်တို့ နံပါတ် System (၁၆) သည် Hexadecimal အရ (10) ဖြစ်သွားပါတယ်။ အောက်ကပုံလေးကို တစ်ချက်ကြည့်ပေးပါ ...



ဒီပုံအရ Hexadecimal တန်ဖိုး (10) သည် ကျွန်တော်တို့ အသုံးပြုနေက Number System အနေနဲ့ စိုရင် တန်ဖိုး (16) နဲ့ တူညီနေမှာ ဖြစ်ပါတယ် ဒီလောက်ဆို Hexadecimal Number System ကို နားလည်သွားလိမ့်မယ်လို ယူဆပါတယ်။



အခုပံ့ထက အဝိုင်း "#" ဆိုင်းလေးတို့သုံးပါတယ် အနောက်က ဂဏန် (၆) ဂုံး မှာ ပထမ နှစ်လုံးက Red (အနီရောင်)၊ အလယ် နှစ်လုံးက Green (အစိမ်းရောင်)၊ နောက်ဆုံးက ဂဏန်း နှစ်လုံးကတော့ Blue (အပြာရောင်) ကို ကိုယ်တားပြုပါတယ်။ တန်ဖိုးအနေနဲ့ က 00 ~ FF အထိ တန်ဖိုးတွေ အတိုးအလျော့လုပ်ပြီး ထည့်သွင်း အသုံးပြုနိုင်ပါတယ်။

အခုဆို Hexadecimal အနေနဲ့ Color တွေကို အသုံးပြုခြင်းကို လေ့လာပြီးသွားပါပြီ။

Color Name တန်ဖိုးအသုံးပြုခြင်း

CSS မှာ Predefined ကြိုတင်ပြီး သတ်မှတ်ထားသော အရောင် (၁၄၇) ရောင်ရှိပါတယ်။ စာဖတ်သူတို့က ရေးသားရမှာ Human Readable ဖြစ်သော အမည်များဖြစ်တဲ့ (ဥပမာ red ဆိုရင် အနီရောင် ပေါ့) အဲလို Color Name (၁၄၇) မျိုးကို အသုံးပြုလိုရပါတယ်။ Hex Color Code ဆိုမဟုတ် RGB method ကိုအသုံးပြုစရာမလိုပဲ တိုက်ရှိက်အသုံးပြုလိုရအောင် CSS က ထောက်ပံ့ပေးပါတယ်။

အသုံးပြုလုပ်ကတော့ eg - p { color: red, background-color: black } ;

တရေးသူလည်း Color Name (၁၄၇) မျိုးလုံးကို သိနေတာ အဂွတ်ရနေတာမျိုးတော့ မဟုတ်ပါဘူး။ စာဖတ်သူတို့ က စိတ်ဝင်စားလို့ သိချင်တယ်ဆိုရင် w3.org ရဲ့ Official website မှာ သွားဖတ်လို့ရပါတယ်။

<https://www.w3.org/TR/css-color-4/#named-colors>

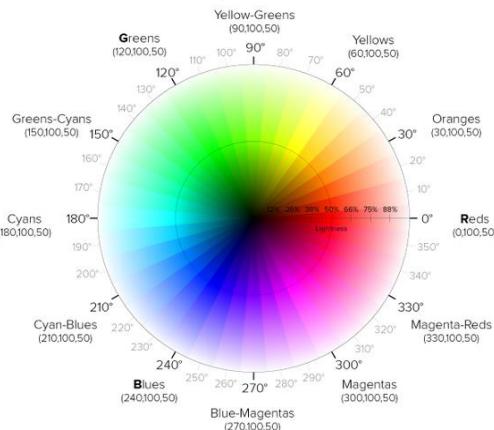
HSL Color တန်ဖိုးအသုံးပြုခြင်း

အရင်ဆုံး HSL ဆိုတာက အရှည်ကောက်အနေနဲ့ Hue , Saturation, Lightness ဆိုပြီး အဓိပ္ပာယ် ရပါတယ်။ ဒီအရောင် စနစ်ကတော့ ပိုပြီး နာလည်လွယ်စေတယ်လို့ ပြောပေါ့မယ် တရေးသူကိုယ်တိုင်က အသုံးနည်းပါတယ်။

H - Hue က Traditional Color Wheel တစ်ခုရဲ့ အရောင်ကို ကိုယ်စားပြုတာပါ။ CSS မှာ hue ကို Color Wheel အရောင်စက်ဝန်း ရဲ့ 0 degree (ဒီဂရိ) မှ 360 degree အတွင်းမှာရှိတဲ့ ထောင့် (Angle) တန်ဖိုးကို ယူပြီး သတ်မှတ်ပေးရပါတယ်။ အခုနားမလည်သေးလည်း ကိစ္စမရှိပါဘူး နောက်တစ်မျက်နှာပုံ Color Wheel အရောင်စက်ဝန်း ပုံနဲ့ တက္က လေ့လာကြည့်ရမှာပါ။ အနီရောင် (Red Color) သည် 0 degree စမှတ်တွင် တည်ရှိပြီး၊ အစိမ်းရောင်မှာ 120 ဒီဂရိဖြစ်ပြီး အပြောသည် 240 ဒီဂရိဖြစ်ပါတယ်။ ဒီဂရိ ထောင့် နံပါတ်ကို ထည့်ပြီး အရောင်ကို သတ်မှတ်တဲ့ ပုံစံပါ။

S - Saturation ဆိုတာက ရာခိုင်နှုန်းအဖြစ် ကိုယ်စားပြုပြီး 0% (မီးခိုးရောင်စကေး) မှ 100% (ထင်ရှားသော အရောင်အဖြစ်) သတ်မှတ်ပေးခြင်းဖြစ်တယ်။ ရာခိုင်နှုန်း နည်းလျင် နည်းနည်း ဖော်တော့နေပြီး၊ ရာခိုင်နှုန်းများလေ ပိုပိုပြီး အရောင်က ပိုတောက်ပပြီး ထင်ထင်ရှားရှားဖြစ်လာတာကို တွေ့ရမှာဖြစ်တယ်။

L - Lightness က တော့ အလင်းပဲ့။ သူက အရောင်ပို တောက်ပလာစေရ ပိုလင်းလာ စေရန် သို့မဟုတ် ပိုမောင်သွားဖို့အတွက် ရွေးချယ်ရာတွင် ကိုယ်တားပြုပါတယ်။ သူ ကိုယ်တိုင်လည်း ရာခိုင်နှစ်းအဖြစ် ဖော်ပြပြီး 0% (လုံးဝအနက်ရောင်) မှ 100% (လုံးဝ အဖြူရောင်) အတွင်းသတ်မှတ်ပေးနိုင်ပါတယ်။ 0% နှင့် 50% အကြားတန်ဖိုးများသည် ပိုမိုနှုန်းသော အမောင်ဘက်ကိုဖြစ်ပေါ်စေပြီး၊ 50% နှင့် 100% အကြားတန်ဖိုးများသည် ပိုမိုလင်းသော အရောင်ဘက်ကိုဖြစ်ပေါ်စေသည်။



အခု Color Wheel (အရောင်စက်ဝန်းကို ကြည့်ကြည့်ပါ) ခုကာ Hue ကိုလေ့လာခဲ့ သလိုပဲ Red အနီရောင်က 0 degree မှာရှိတယ်၊ Green အထိမ်းက 120 degree ဖြစ်ပြီး၊ Blue အပြာက 240 degree မှာရှိတာကိုတွေ့ရှိရမှာပါ။

ဆိုတော့ CSS ရေးသားနည်းနဲ့ ဘယ်လို့ အသုံးပြုရမလဲဆိုတာ လေ့လာကြည့်ရအောင်။

eg - hsl(0, 100%, 50%)

$h = 0$, Hue က 0 degree မှာရှိသော အရောင် Red ဆိုတဲ့အတွက် အနီရောင်ပါ။

$s = 100\%$, Saturation က တော့ Gray Scale မီးခိုးရောင်း က 0% ဖြစ်ပြီး 100% အဖြစ် သတ်မှတ်ထားတဲ့အတွက် အထင်ရှုးဆုံးဖြစ်အောင်ပြုလုပ်ထားတာပါ။

$l = 50\%$, Lightness အစိုင်းမှာက 50% နှင့် 100% အကြားတန်ဖိုးများသည် ပိုမိုလင်းသော အရောင်ဘက်ကိုဖြစ်စေသောကြောင့် အလင်းရဲ့ အစ 50% ကိုပေးထားလိုက်ပြီး ဖြစ်တယ်။

စာဖတ်သူတို့ အရောင် Angle ဖြစ်တဲ့ Hue ရဲ့ ထောင့်နဲ့ပါတ် ရပြီဆိုတာနဲ့ အလင်းအမောင်ဖြစ်တဲ့ s, l တို့ကို အလျော့အတင်းလုပ်ပြီး အလွယ်တကူပြောင်းလဲနိုင်သောကြောင့် Designer များက HSL အရောင် စနစ် သုံးလေ့ရှိကြပါတယ်။ စာရေးသူ ကိုယ်တိုင်ကတော့ လုံးဝကို မသုံးဖြစ်ပါ။

CSS Opacity ကိုလေ့လာခြင်း

Opacity ကိုလေ့လာမယ်ဆုံး နားလည်အောင်ပြောရရင် ဖောက်မြင်အောင် လုပ်လိုက်တာပဲပျော်။ မှန်တွေလို့ပေါ့ ဟိုဘက်ဒီဘက်ကို မြင်နိုင်သွားတာကို ဆိုလိုတာပါ။

ပထမဦးစွာ ကျွန်တော်တို့ New Folder (ဖိုလ်ဒီအသစ်) တစ်ခု "18_css_color_opacity" ဆိုပြီးတည်ဆောက်လိုက်ပါမယ်။ ပြီးရင် အဲ ဖိုလ်ဒီအတွင်းမှာ "index.html" နဲ့ "style.css" ဆိုပြီး HTML နဲ့ CSS ဖိုင်(J) ခုတည်ဆောက်လိုက်ပါမယ်။

index.html အတွင်းမှာ အောက်က Gist Code ကို ကူးယူလိုက်ပါ။

<https://gist.github.com/lwinmoeapaing/379ceb387e1142eddb20ddc5468c4ec7>

```
● ● ● By Lwin Moe Paing 18_css_color_opacity/index.html

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<title>CSS Color | Lwin Moe Paing</title>
<link rel="stylesheet" href="style.css" type="text/css" />
</head>
<body>
<div class="container pre-size">
<div class="square pre-size box-1">
<p class="text-white">Box 1</p>
</div>
</div>
<div class="container pre-size">
<div class="square pre-size box-2">
<p class="text-white">Box 2</p>
</div>
</div>
<div class="container pre-size">
<div class="square pre-size box-3">
<p class="text-white">Box 3</p>
</div>
</div>
</body>
</html>
```

style.css အတွက် အောက်က Gist Code ကို ကူးယူလိုက်ပါ။

<https://gist.github.com/lwinmoeapaing/054cfb0f015e7fd4fa6fb3fd239c8008>

```
18_css_color_opacity/style.css By Lwin Moe Paing


```

အောက်ကပုံအတိုင်း Browser တွင်ပေါ်နေပါလိမ့်မယ် ..

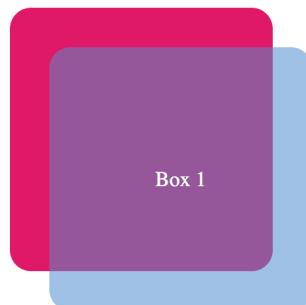


ကျွန်တော်တို့ ကျွန်တဲ့ CSS style Rule တွေကို ခကဗော်ထားလိုက်ပါ။ အရင်ဆုံး Box 1 အတွက် .box-1 { background-color: rgb(55, 130, 203); } ရေးထားတာကို မြင်ရမှာပါ။ သူက Box 1 အတွက် နောက်ခံအရောင်ကို အပြာရောင်အဖြစ်သတ်မှတ်ထားခြင်းဖြစ်ပါတယ်။

စာဖတ်သူတို့နဲ့ အတူတူ ကျွန်တော်တို့ အဲဒီ Box ကို ဖောက်ပြီးမြင်ရအောင် Opacity ပေးကြည့်ပါမယ်။ အောက်ပါ Code အတိုင်း စာဖတ်သူကလည်း ပြောင်းလဲရေးသားပေးပါ။

```
.box-1 {
    background-color: rgba(55, 130, 203, 0.5);
}
```

အဓိပ္ပာယ်က rgb ကတော့ ကျွန်တော်တို့ လေ့လာထားပြီးဖြစ်တဲ့ Red,Green,Blue ကို ညွှန်းဆိုပါတယ် အနောက်က A ကတော့ Alpha ပါ သူရဲ့ တန်ဖိုးက 0 ကနေ 1 ကြားထဲ တန်ဖိုးသတ်မှတ်ပေးနိုင်ပါတယ်။ 1 က တန်ဖိုးအကြိုးဆုံးဖြစ်ပြီး ဖောက်ထွင်းမြင်နိုင်မှု ပိတ်အားအကြိုးဆုံး လို့ဆိုလိုတာပါ။ တန်ဖိုး (0) ထားလိုက်ရင် အားလုံး ဖောက်ထွင်းမြင်ပိတ်အားက သူည့် ဖြစ်သွားသောကြောင့် ဖောက်ထွင်းအား အရမ်းကြီးသွားမှာဖြစ်ပါတယ်။ ဒါဆိုရင် Background နောက်ခံအရောင်က ပျောက်သွားပါလိမ့်မယ်။ စာဖတ်သူတို့က (0) ကနေ (1) ကြားအတိုးအလျော့လုပ်ကာ စမ်းသပ်နိုင်ပါသည်။ အခုလက်ရှိမှာ ဝေးထော့ 0.5 ထားထားသောကြောင့် အောက်က ပုံအတိုင်းဖြစ်နေပါလိမ့်မယ်။



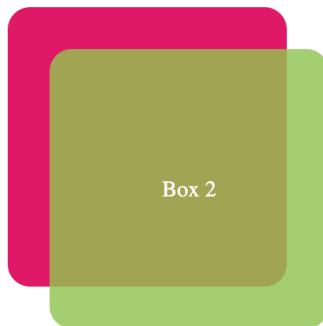
0.5 ပေးထားသော Box 1 ပုံကိုမြင်ရခြင်းဖြစ်ပါတယ်။ နောက် တစ်မျက်နှာမှာ Box 2 အတွက် Hexadecimal Code ကို Opacity ဖြစ်အောင် လုပ်ကြည့်ကြပါမယ်။

Box 2 အတွက် .box-2 { background-color: #8bc34a; } ရေးထားတာကို မြင်ရမှာပါ။ သူက Box 2 အတွက် နောက်ခံအရောင်ကို အစိမ်းရောင်အဖြစ် သတ်မှတ်ထားခြင်းဖြစ်ပါတယ်။ Hex Code Color ကို အောက်ကအတိုင်းပြောင်းရေးလိုက်ပါမယ်

```
.box-2 {  
    background-color: #8bc34acc;  
}
```

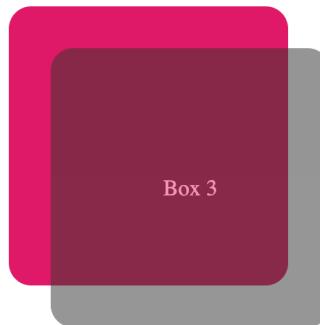
အဓိပါယ်က "#" sign နောက်က 8bc34a ဆိုတဲ့ ဂကန်းမြောက်လုံးက Red, Green, Blue ကို ကိုယ်စားပြုပါတယ်။ အနောက်က ထပ်တိုးလာတဲ့ ဂကန်းနှစ်လုံးကတော့ Alpha ဖြစ်တဲ့ Opacity အတိုးအလျော့တန်ဖိုးပဲဖြစ်ပါတယ်။

(ff) က တန်ဖိုးအကြီးဆုံးဖြစ်ပြီး ဖောက်ထွင်းမြင်နိုင်မှပိတ်အားအကြီးဆုံး လိုအပိုလိုတာ ဖြစ်ပါတယ်။ (00) တန်ဖိုးကတော့အနည်းဆုံးပါ။ စာဖတ်သူတို့က (00) မှ (ff) ကြား အတိုးအလျော့လုပ်ကာ စမ်းသပ်နိုင်ပါသည်။ အခုလက်ရှိမှာတော့ (cc) ထားထား သောကြောင့် အောက်က ပုံအတိုင်းဖြစ်နေပါလိမ့်မယ်။



Box 3 အတွက် ကျွန်ုတ်တို့ အရောင်တွေတို့မလုပ်တော့ပဲ Element တစ်ခုလုံးကို Opacity ဖောက်ခေါ်ပေါင်တဲ့အပါ အောက်က Code အတိုင်းရေးလိုက်လို့ရပါတယ်။

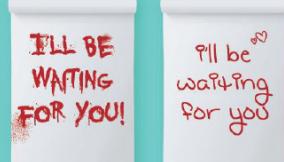
```
.box-3 {  
    background-color: #2d2d2d;  
    opacity: 0.5;  
}
```



Box 3 အတွက် opacity (Property) ကိုသုံးလိုက်ခြင်းအားဖြင့် element တစ်ခုလုံးကို သက်ရောက်စေမှာဖြစ်ပါတယ်။ သူက Background Color ဖြစ်တဲ့ နောက်ခံ အရောင် သာမက Content ဖြစ်တဲ့ "Box 3" ဆိုတဲ့ ဘသားပါ ဖောက်ဝင်မှုကို သက်ရောက်နေမှာ ကို သတိပြုစေချင်ပါတယ်။ Property (opacity) ရဲ့ တန်ဖိုးကဗျည်း (0) ကနေ (1) ကြား သတ်မှတ်ပြီး အသုံးပြုနိုင်ပါတယ်။ ဒီလောက်ဆို ကျွန်ုတ်တို့ Opacity ဖြစ်တဲ့ Transparency ဖောက်ထွင်းမြင်အောင် ရေးသားနည်းကို လေ့လာပြီးဖြစ်ပါတယ်။

အခု Code အားလုံးကို အောက်က Codepen မှာ သွားရောက်လေ့လာနိုင်ပါတယ်

<https://codepen.io/lwinmoepaing/pen/gOQGPxr>



FONT CHOICE **MATTERS**

Fonts တွေရွေးချယ်တာအရေးကြီးတယ်

Typo တွေက အရမ်းအရေးကြီးပါတယ်။

I'll be Wating For you ဆိုတဲ့ စာသားကို ကြည့်လိုက်ပါ
ဘယ်ဘက်တစ်ခုကတော့
သင့်ကိုသတ်တော့မယ့်အတိုင်းပါပဲ ...

CSS နှင့် Fonts များအကြောင်း

Font Family များသို့ခရီးထွက်ခြင်း

Font (ဖောင်) တစ်ခုဆိတ်တာ သီးခြားသတ်မှတ်ထားတဲ့ Typeface ကိုယူသုံးထားသော Fonts Style Group (ဖောင်စတိုင်အုပ်စု) လိုအလွယ်ပြောလို့ရပါတယ်။ Font တစ်ခုမှာ မတူညီတဲ့ Fonts style တွေပါဝင်ပါတယ် ဥပမာ - Regular Font, Bold Font, Light Font ဆိုပြီး ပါဝင်ပါတယ်။ သို့ပေါ်မယ် အဲဖောင်တွေအားလုံးက တူညီတဲ့ Design Concept (အယူအဆ) Typeface တစ်ခုရဲ့အောက်မှာပဲ ဖန်တီးထားတယ်လို့ ဆိုလိုခြင်းဖြစ်တယ်။

Font Family ကတော့ Fonts တွေကို Group (ဖောင်အုပ်စုတဲ့) ပုံစံနဲ့ အသုံးပြုတာကို ခေါ်တာဖြစ်ပါတယ်။ Font Family ကို Type (အမျိုးအစား) ကိုထပ်ခွဲကြည့်မယ်ဆို အဓိကအားဖြင့် Generic Font Family နဲ့ Specific Font Family ဆိုပြီး Type (J) မျိုးကို တွေရပါမယ်။

Specific Font Family အမျိုးအစားမှာ Designer က ဒီ Design အတွက် ဒီ Font တစ်ခုကိုပဲ အသုံးပြုမယ်။ တိတိကျကျ ဒါကိုပဲသုံးတဲ့အခါမျိုးမှာသုံးလေ့ရှိပါတယ်။

Generic Font Family ကတော့ Similar (ပုံစံဆင်တူတဲ့) Typeface တွေသုံးထားတဲ့ Fonts တွေကို အုပ်စုပြီးသုံးလေ့ရှိပါတယ်။ Common Generic Fonts တွေကတော့ serif, sans-serif, monospace အစရှိတဲ့ ဖောင်တွေဖြစ်ပါတယ် အခြားအများကိုကျန်ပါသေးတယ်။

စာဖတ်သူတို့က Terms (စာကားလုံးအသစ်) တွေကြောင့် နည်းနည်းအခက်အခဲရှိနိုင်ပါတယ်။ ကျွန်ုတ်တော်တို့ Web Development လောကမှာတော့ Generic Font Family ကိုပဲ အသုံးများပါတယ်။

Genric Font Family က ဘာတွေလုပ်ပေးလဲဆိုရင် ကျွန်ုတ်တော်တို့ Website ကို အသုံးပြုတဲ့အခါ အသုံးပြုသူရဲ့စက်ထဲမှာ ကျွန်ုတ်တော်တို့ သတ်မှတ်ထားသော Fonts တွေမရှိနိုင်တာဖြစ်နိုင်တယ် (သို့မဟုတ်) Internet connection တစ်ခုရဲ့ကြောင့် Fonts တွေကိုယူလို့မရတဲ့တာမျိုးဖြစ်တတ်ပါတယ်။ အဲလိုအခြေအနေမျိုးမှာ သူက Fallback အနေနဲ့ အလားတူ (ပုံစံဆင်တူတဲ့) အသုံးပြုသူစက်ထဲက Font ကိုယူပြလိုက်ပါတယ်။ ဒီတော့ ကျွန်ုတ်တို့ သုံးထားတဲ့ Font မပေါ်ပေါ့မယ့်လည်း Design အရတော့ သိသိသာသာ ပြောင်းလဲမှုရှိမှာမဟုတ်တဲ့ အားသာချက်ကို ရနိုင်ပါတယ်။

- ကျွန်ုတ်တို့ နောက်တစ်မျက်နှာမှာ Typeface တွေအကြောင်းလေ့လာကြမယ်။

CSS နှင့် Typeface များအကြောင်း

Font (ဖောင်) တစ်ခုရှိတာ သီးချားသတ်မှတ်ထားတဲ့ Typeface ကိုယူသုံးထားသော Fonts Style Group (ဖောင်စတိုင်အပ်စု) ရှိပြီးလေ့လာခဲ့ပါတယ်။ ပြီးတော့ တူညီတဲ့ Design Concept (အယူအဆ) Typeface တစ်ခုရဲ့အောက်မှာပဲ တည်ရှုတယ်ပေါ့။ အောက်က Google က Roboto Font ကိုလေ့လာကြည့်ရအောင်။

Roboto Font

| Typeface | Font Style |
|----------|--|
| Roboto | Roboto Light Roboto Thin Roboto Regular Roboto Medium Roboto Bold Italic Roboto Black Italic Roboto Light Italic Roboto Thin Italic Roboto Italic Roboto Medium Italic Roboto Bold Italic Roboto Black Italic |

ပုံစံဘဏ္ဍာရင် Roboto ဆိုတဲ့ Font လေးပါပဲ သို့ပေမယ် သူမှာ Style မျိုးစုံပါပါသေး တယ်။ Font Style များကလည်း Roboto ဆိုတဲ့ Typeface အောက်ကနေ သူသတ်မှတ်ထားတဲ့ Design Concept နဲ့ Characteristics ကိုသွေဖယ်မသွားတဲ့ ပုံစံနဲ့စွဲ့စည်းထားတာပါ။

ဒါနို ကျွန်ုတော်တို့ Common ယေဘုယျအားဖြင့် Web Development နယ်ပယ်မှာ အသုံးများတဲ့ Fonts တွေကိုလေ့လာကြည့်ပါမယ်။ ဤတော့ သူတို့ရဲ့ Typeface ကို အခြေခံပြီး အကြော်း Fonts များစွာကိုလည်း Generic Font Family မှာ ထည့်သွင်း အသုံးပြုကြပုံကိုလည်း လေ့လာသွားပါမယ်။

Common ဖြစ်ပြီး အသုံးများတဲ့ Generic Fonts တွေကိုအောက်မှ ဖော်ပြထားပါတယ်။

serif



Serif Font

sans-serif

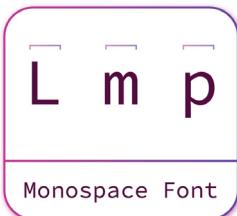


Sans-Serif Font

အလွယ်ဆုံးနည်းနဲ့ ဆိုရင် စကားလုံး
တစ်လုံးရဲ့ ခြေထောက်၊ ခေါင်းထိပ်လေး
တွေမှာ အချင့်လေးတွေပါပြီး အသေးစိပ်
ပြုလုပ်ထားတဲ့ font တွေကို Serif Font တွေ
လိုက်ပါတယ်။ အများစုံက Header ဖြစ်တဲ့
Webpage ရဲ့ ခေါင်းစီးပိုင်းတွေရေးတဲ့အခါ
Serif Font ကိုအသုံးများပါတယ်။

Font Style အရ Detail (အသေးစိပ်) မကျ
ပေမယ့် Website ရဲ့ စာသားတွေမှာ အသုံး
များတဲ့ Font တစ်ခုဖြစ်ပါတယ်။ ရုံရုံး
ရှင်းရှင်း ဖြစ်ပြီး စာဖတ်တဲ့အခါ ရှင်းလင်းတဲ့
အတွက် Blog Contents တွေ (သို့) စာပိုဒ်
အရှည်တွေမှာ အသုံးပြုပါတယ်။ ဒါကြောင့်
ဖတ်ရလွယ်ပြီး ရှင်းလင်းတဲ့ Font တွေ
ကတော့ Sans Serif တွေပဲဖြစ်ပါတယ်။

monospace



Monospace Font

အပေါ်မှာ မျဉ်းလိုင်းတားထားတဲ့ အကျယ် Width
Space တွေက အကုန်တူညီနေတာ က Mono-
spaced Font တွေရဲ့ ထူးမြားချက်တစ်ချက်ပဲဖြစ်
ပါတယ်။ Developer တွေရဲ့ IDE တွေနဲ့ Code
Editor တွေမှာအသုံးအများဆုံး Font တစ်ခုပါပဲ။
ဘာလိုက်လည်းကောင်းမြတ်ရလွယ်ကြုံး ပြန်လည်ပြပြင်ရ^၁
တဲ့ခါ ညီညာပြီး အတန်းလိုက်စိတ္တားသလို အမြင်
လွယ်ကူတာကြောင့်ဖြစ်ပါတယ်။ ပြီးတော့ Terminal
မှာလည်း ဒီ Font တွေအသုံးများပါတယ်။

FANTACY



FANTACY Font

Fantacy Font အမျိုးအစားကတော့ Standard
တွေကို လွှန်ဆန်ပြီး Hand Drawn Font မျိုး
လည်းဖြစ်နိုင်တယ်။ စိတ်ကူးယဉ်ဆန်ပြီး ဆန်း
ဆန်းပြားပြားမျိုးတွေ ဖြစ်ပါတယ်။ အများ
အားဖြင့် Title တွေ ခေါင်းစီးပိုင်းတွေမှာ အသုံး
များပြီး အမြား Website တွေနဲ့မတူတဲ့ ဥပမာ
Gaming Website တွေမှာတွေ ရတတ်ပါတယ်။

CSS နှင့် Fonts များကို အသုံးပြုခြင်း

Fonts Family | Typeface တွေကို ကျွန်တော်တို့ အကျဉ်းချုပ်လေ့လာပြီးတဲ့ နောက်မှာ လက်တွေ၊ Coding နဲ့ CSS မှာဘယ်လို့ Apply လုပ်မလဲ လေ့လာသွားကြပါမယ်။

ပထမဦးစွာ ကျွန်တော်တို့ New Folder (ဖိုလ်ဒါအသစ်) တစ်ခု “19_css_font” ဆိုပြီးတည်ဆောက်လိုက်ပါမယ်။ ပြီးရင် အဲ ဖိုလ်ဒါအတွင်းမှာ ထပ်ပြီး နောက်ထပ် ဖိုလ်ဒါအသစ် “1_font_family” ဆိုပြီး တည်ဆောက်လိုက်ပါမယ်။ “1_font_family” ဆိုတဲ့ အတွင်းထဲမှာ “index.html” နဲ့ “style.css” ဆိုပြီး HTML နဲ့ CSS ဖိုင် (J) ခုတည်ဆောက်လိုက်ပါမယ်။

index.html အတွက် Gist Code ကို ကူးယူလိုက်ပါ။

<https://gist.github.com/lwinmoepaing/f6a45b68843f76030cbf01660f7ea68f>



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Font Family | Lwin Moe Paing</title>
    <link rel="stylesheet" href="style.css" type="text/css" />
</head>
<body>
    <h1>H1 Title - Tag</h1>
    <p>
        Lorem ipsum dolor sit, amet consectetur adipisicing elit. Atque cupiditate
    </p>

    <h2>H2 Title - Tag</h2>
    <p>
        Lorem ipsum dolor sit, amet consectetur adipisicing elit. Atque cupiditate
    </p>
</body>
</html>
```

style.css အတွက် Gist Code ကို ကူးယူလိုက်ပါ။

<https://gist.github.com/lwinmoepaing/597463b885189a48b3637d84a75fca3d>



```
h1 {
    font-family: "Palatino Linotype", "Book Antiqua", Palatino, serif;
}
p {
    font-family: Arial, Helvetica, sans-serif;
}
```

တာဖတ်သူတို့က Browser တွင်စမ်းကြည့်တဲ့အခါ h1 အတွက် serif ဖောင့်အမျိုးအစားနဲ့
ပေါ်နေမှာဖြစ်ပြီး။ p အတွက်က sans-serif ဖောင့်အမျိုးအစားနဲ့
ပေါ်နေမှာပါ။ CSS
property ဖြစ်တဲ့ font-family အရေးအသားကို လေ့လာကြည့်ရအောင်။ <h1> element
အတွက်ကို အောက်ပါအတိုင်း ရေးသားထားသည်။

font-family: "Palatino Linotype", "Book Antiqua", Palatino, serif;

h1 element အတွက် ပထမ Palatino Linotype ဆိုတဲ့ Fonts ကို ပေးထားပါတယ်။ သို့
ပေါ့မယ် အနောက်က "Book Antiqua", Palatino, serif; ဆိုတာတွေက Fallback Fonts
တွေဖြစ်ပါတယ်။ ဆိုလိုချင်တာက Palatino Linotype ဆိုတဲ့ ဖောင့်က အသုံးပြုဘူး။
User တွေရဲ့ Browser စက်မှာ Available (ရှိနေပြီးသား) ဆိုရင်ပြုမယ် မရှိခဲ့ဘူးဆိုရင်
အနောက်က ဖောင့်တွေကို ဆက်ပြီးတော့ အသုံးပြုပေးမှာပါ။ ဥပမာ "Palatino Linotype"
မရှိရင် "Book Antiqua" တို့သုံးမယ်၊ "Book Antiqua" မရှိရင် Palatino ကိုသုံးပါ
မယ်၊ Palatino မရှိခဲ့ရင်တော့ serif ဖောင့်ကို အသုံးပြုပါမယ်ဆိုပြီးဖြစ်ပါတယ်။

Default System Fonts တွေကို အောက်က List မှာ ဖော်ပြထားပါတယ်။ တာဖတ်သူတို့
စက်တွေမှာလည်း Common အနေနဲ့ ဘုံပါတဲ့ Fonts တွေဖြစ်တဲ့အတွက် font-family
နေရာမှာ အောက်က ဖောင့်တွေပြောင်းပြီးတော့ စမ်းသပ်အသုံးပြနိုင်ပါတယ်။ ဖြစ်နိုင်
ရင် Spacebar မြားထားတဲ့ Fonts တွေကို "" Double Quote အသုံးပြုပြီးရေးပေးပါ။

Arial, Helvetica, Sans-Serif;

Arial Black, Gadget, Sans-Serif;

Comic Sans MS, Textile, Cursive;

Courier New, Courier, Monospace;

Georgia, Times New Roman, Times, Serif;

Impact, Charcoal, Sans-Serif;

Lucida Console, Monaco, Monospace;

Lucida Sans Unicode, Lucida Grande, Sans-Serif;

Palatino Linotype, Book Antiqua, Palatino, Serif;

Tahoma, Geneva, Sans-Serif;

Times New Roman, Serif;

Trebuchet MS, Helvetica, Sans-Serif;

Verdana, Geneva, Sans-Serif;

MS Sans Serif, Geneva, Sans-Serif;

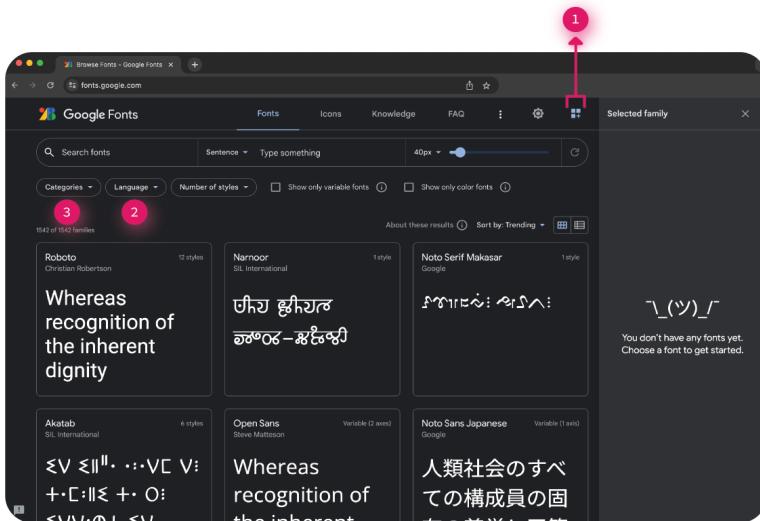
MS Serif, New York, Serif;

CDN fonts အသုံးပြုခြင်း

CDN က တော့ Content Delivery Network ကိုဆိုလိုတာဖြစ်ပါတယ်။ ဒီတစ်ခါမှာတော့ ကျွန်တော်တို့ System Fonts တွေမဟုတ်ဘဲ အသုံးပြုသူတွေရဲ့ စက်ထဲမှာမရှိတဲ့ Google Font တွေကို အသုံးပြုမှာပါ။ Google Font API Service ကမောင် တိုက်ရှိက်ယူသုံးတာဖြစ်ပါတယ်။ အသုံးပြုရောမှာ နည်း (၂) မျိုးနဲ့အသုံးပြုလိုရပါတယ် ပထမတစ်ခုက <link> element နဲ့ ချိန်ဆက်အသုံးပြုတာဖြစ်ပြီး၊ CSS ဖိုင်ထဲ မှာပဲ တစ်ခါတည်း import လုပ်ကာ အသုံးပြုခြင်းဖြစ်ပါတယ်။

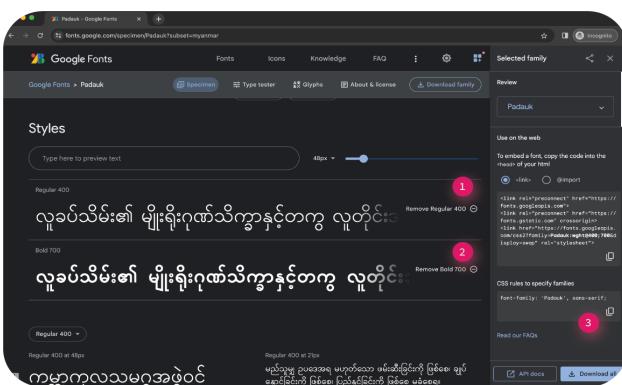
အရင်ဆုံးကျွန်တော်တို့ Google Fonts တွေရှိရာ အောက်ကလင့်ကို ဝင်လိုက်ပါ။

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



ပုံထဲကအတိုင်း ညာဘက် Selected Family ဆိုတဲ့ Side Bar မပေါ်ပါက အညွှန်း (၁) တွင်ပြထားသော Icon လေးကိုနှိပ်ပေးပါ။ အညွှန်း (၂) နေရာက Language ကို ရွေးချယ်ပြီး စာဖတ်သူတို့ ကြိုက်နှစ်သက်ရာ Langague ဘာသာစကားများကိုရှာနိုင်ပါတယ်။ အညွှန်း (၃) အရ Serif, Sans-Serif တို့ Monospace အစဉ်တဲ့ Font Category တွေကို ရွေးချယ်နှင့်ပါတယ်။

တာရေးသူကတော့ အညွှန်း (၂) ထဲမှာ Myanmar ကိုရွေးချယ်လိုက်ပါတယ် ဒါဆိုရင် တော့ တွေ့လာရမယ့် Fonts တွေကတော့ Padauk, Noto Sans Myanmar, Noto Serif Myanmar ဆိုပြီး တွေ့ရပါတယ်။ Padauk Font ကိုနှိပ်လိုက်ပါက နောက်တစ်မျက်နှာ က ပုံအတိုင်းဖြစ်သွားသည်ကို ပြင်ပေါ်မယ်။



ပုံထဲကအတိုင်း Padauk ဖောင့်ကို နှိပ်ပြီးပါက အညွှန်း (၁) နဲ့ (၂) တွင် Select မှတ်လိုက်ပါက။ ညာဘက်က Side Bar တွင် Use on the web ဆိုပြီး <link> နဲ့ ချုပ်ကာ အသုံးပြုချင်း (သို့မဟုတ်) @import method နဲ့အသုံးပြုခြင်းကို ရွေးချယ်နှင့်သွားမှာ ဖြစ်ပါတယ်။ အညွှန်း (၃) ထဲကအတိုင်း <link> နဲ့ချိတ်ဆက်ပြီး သုံးချင်ပါက အပေါ်က <link> ဆိုတဲ့ Select Box လေးကိုရွေးချယ်ပြီး <link rel="..."> ကနေစပြီး ကူးကာ <head> element အတွင်း ယူသုံးပေးရမှာ ဖြစ်ပါတယ်။ @import နဲ့ အသုံးပြုချင်ရင် တော့ @import ဆိုတဲ့ Select Box လေးကိုရွေးချယ်ပြီး အတဲ့ရှိ Style Rule ကို style.css ထဲမှာ ကူးယူပြီး အသုံးပြုနိုင်ပါတယ်။

"19_css_font" ဖိုလ်ဒါအတွင်းမှာ ဖိုလ်ဒါအသစ် "2_cdn_fonts" ဆိုပြီး တည်ဆောက်လိုက်ပါမယ်။ "2_cdn_fonts" အတွင်းထဲမှာ "index.html" နဲ့ "style.css" ဆိုပြီး HTML နဲ့ CSS ဖိုင် (၂) ခုတည်ဆောက်လိုက်ပါမယ်။

index.html အတွက် အောက်ပါ Gist Code ကို ကူးယူလိုက်ပါ။

<https://gist.github.com/lwinmoeapaing/0797f0c4ec19e3b8a3cf1b02913cb72>

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8" />
<meta name="viewport" content="width=device-width, initial-scale=1.0" />
<title>Linking Fonts | Lwin Moe Paing</title>
<!-- Google Font -->
<link rel="preconnect" href="https://fonts.googleapis.com" />
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin />
<link href="https://fonts.googleapis.com/css2?family=Padauk:wght@400;700&display=swap" rel="stylesheet" />
<!-- our style.css -->
<link rel="stylesheet" href="style.css" />
</head>
<body>
<h1>Padauk</h1>

<p>ဒါကေတော့ ပိုတော်မောင်ကို အသုံးပြုပြီး ရေးသားထားခြင်းပြုခြင်းပါတယ်.</p>
</body>
</html>
```

style.css အတွက် အောက်ပါ Gist Code ကို ကူးယူလိုက်ပါ။

<https://gist.github.com/lwinmoepaing/5ac8f7f8559da35cc30d0cdc9010f4cc>



```
● ● ● 19_css_fonts/1_cdn_fonts/style.css
p {
    font-family: "Padauk", sans-serif;
}
```

အခုဆိုရင် ဘဖတ်သူတို့ Browser တွင် ဖွင့်ကြည့်ပါက ကျွန်တော်တို့ <p> element အတွင်စားသားတွေက Padauk (ပိတေက်) ဖောင့် အနေနဲ့ ဖော်ပြုသွားမှာဖြစ်ပါတယ်။ style.css အတွင်း @import နည်းကို အောက်တွင်ထပ်ဖော်ပြုသွားပါမယ်။ index.html အတွင်း အောက်က <!-- Google Font --> ကနေစပြီး ပြန်ဖျက်လိုက်ပါ။

```
<!-- Google Font -->
<link rel="preload" href="https://fonts.googleapis.com" />
<link rel="preload" href="https://fonts.gstatic.com" crossorigin />
<link href="https://fonts.googleapis.com/css2?family=Padauk:wght@400;700&display=swap" rel="stylesheet">
/>
```

သို့ပေမယ် style.css ကိုချိတ်ထားတဲ့ <link> element ကိုတော့ ချွန်ထားခဲ့ပေးပါဉြီး။ ဒြိုးနောက် style.css အတွင်း အောက်ပါ ရေးသားနည်းအတိုင်း ရေးသားလို့ရပါတယ်။



```
● ● ● 19_css_fonts/1_cdn_fonts/style.css
@import url("https://fonts.googleapis.com/css2?family=Padauk:wght@400;700&display=swap");

p {
    font-family: "Padauk", sans-serif;
}
```

style.css အတွက် @import နည်းကို အောက်ပါ Gist Code တွင် ကူးယူနိုင်ပါတယ်။

<https://gist.github.com/lwinmoepaing/24bc394a91563c65149dc77c4129bf61>

မှတ်ချက်။ ။ ။ CSS အတွင်း @import စကားလုံးသည် ပြင်ပ Resources Files တွေ သို့မဟုတ် ပြင်ပ CSS ဖိုင်တွေကို ချိတ်တဲ့အခါသုံးနိုင်ပါတယ်။ သို့ပေါ့မယ် ပို့ကောင်းသော performance and compatibility အရ <link> element ဖြင့်သာသုံးရန်အကြံ့ပေးပါတယ်။

Custom fonts များအသုံးပြုခြင်း

လက်တွေ၊ လုပ်ငန်းခွင့်မှာတော့ Company က ပေးတဲ့ Custom Fonts သို့မဟုတ် သူတို့ ဝယ်ယူထားတဲ့ Premium Fonts တွေကို အသုံးပြုလေ့ရှိကြပါတယ်။ ဒါနဲ့ရင် Google ပေါ်မှာလည်းရှိမှာမဟုတ်သလို အသုံးပြုဘူး (User) တွေရဲ့ စက်တွေမှာလဲ ရှိမယ်ဆိုတာ မကြမ်းသေပါဘူး။ ဒီဂို့ အကြောအနေမျိုးမှာ ကျွန်ုတ်တော်တို့က Custom Font အသုံးပြုနည်းနဲ့ CSS မှာ ချိတ်ဆက်ပေးနိုင်ပါတယ်။

ဥပမာ စာဖတ်သူနဲ့ ကျွန်ုတ်တော်တို့ ရုံးတစ်ခုမှာ အတူတူတွဲလုပ်နေကြတယ်ဆိုပါစို့။ ရုံးမှာ Boss က idea ရှိလာပါတယ်။ Halloween ကာလအတွက် Web Page မှာရှိတဲ့ ခေါင်းစဉ်တွေအားလုံးကို အောက်က Zombie Holocaust Font ကို ပြောင်းစေချင်တာပါ။ အောက် Team မှာရှိတဲ့ Designer တွေက ကျွန်ုတ်တော်ကို Font ကို ttf (ဖိုင်) အနေနဲ့ ပို့လိုက်ပါတယ်။ အခု စာဖတ်သူနဲ့ ကျွန်ုတ်တော်က ဒီဇော်ကို ပြောင်းပေးရတော့မှာပါ။

ZOMBIE HOLOCAUST

အရင်ဆုံး စာဖတ်သူတို့က ကျွန်ုတ်တော်နဲ့ အတူတူ “19_css_font” ဖိုလ်ဒါအတွင်းမှာ ဖိုလ်ဒါအသုစ် “3_custom_fontface” ဆိုပြီး တည်ဆောက်လိုက်ပါမယ်။ “3_custom_fontface” အတွင်းထဲမှာ “index.html” နဲ့ “style.css” ဆိုပြီး HTML နဲ့ CSS ဖိုင် (J) ခု တည်ဆောက်လိုက်ပါမယ်။

ပြီးရင်အောက်မှာ Font Link ကို ကျွန်ုတ်တော်ချေပေးထားပါတယ်။ စာဖယ်သူတို့က သွားပြီး Download (ခေါင်း) ပေးပါ။

<https://drive.google.com/file/d/1mnNLMDhnQhHGrO-cSBDSIqKlgDVDd9L6/view?usp=sharing>

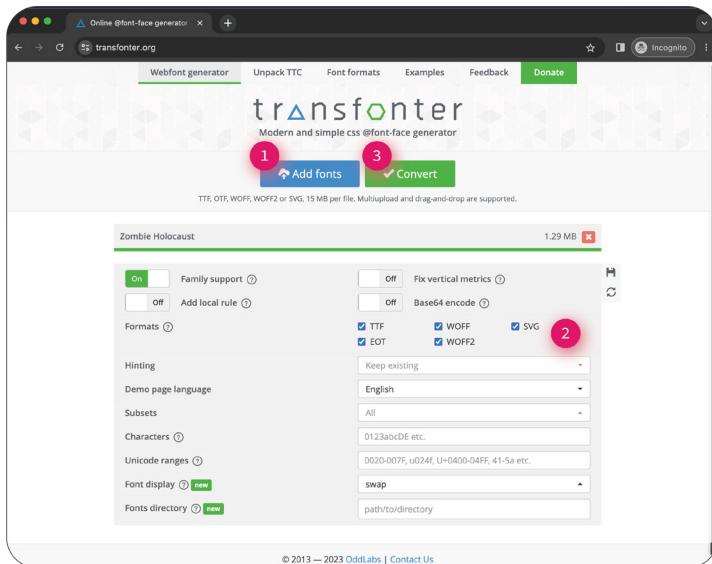
ပုံမှန်အားဖြင့် Font တွေဟာ Format (သို့) extension (ဖိုင်အမျိုးအစား) တွေအနေနဲ့ .ttf, .otf, .woff, .woff2, .svg အစရှိတဲ့ Format တွေနဲ့ အသုံးပြုလေ့ရှိပါတယ်။ Browser အမျိုးမျိုးကလည်း မတူညီတဲ့ Format တွေကို Support ပုံခြင်းလည်းကွာပါတယ်။ အောက်ကလယားကို ကြည့်ကြည့်ပါ။

| Browser | eot | woff | ttf/otf | svg |
|--------------|-----|------|---------|-----|
| Chrome (all) | | | | ● |
| Chrome 6+ | | ● | ● | ● |
| Firefox 3.5 | | | | |
| Firefox 3.6+ | | ● | ● | |
| IE 5 - 8 | ● | | | |
| IE 9+ | ● | ● | ● | |
| Opera 10+ | | | ● | ● |
| Safari 3.1+ | | | ● | ● |
| iOS < 4.2 | | | ● | ● |
| iOS > 4.2 | | | ● | ● |

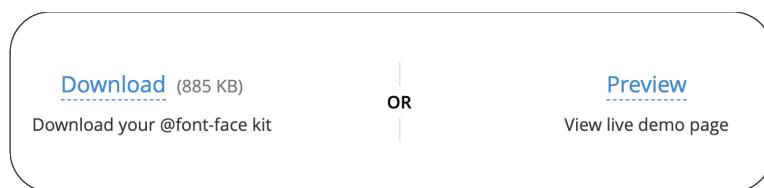
အဲတော့ ဘယ်လို အဲအမျိုးအစားတွေထဲက ဘယ်လို Format မျိုးနဲ့ပဲရလာလာ ကျွန်တော်တို့ ttf (သို့) ttf ဖိုင်အမျိုးအစားကတော့ အသုံးပြုရ အလှယ်ဆုံးပါပဲ။ Browser တော်တော်များများလည်း Support တဲ့အတွက်ကြောင့်ပါ။ ဒါကို စာဖတ်သူတို့ က Custom Font သုံးမယ်ဆုံး သတိထားပေးရပါမယ်။

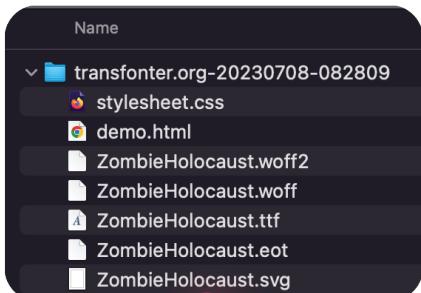
ကံကောင်းတာက ကျွန်တော်တို့ Modern ခေတ်မှာ Fontface Generator သုံးပြီး Browser အားလုံးသုံးလို့ရအောင် ttf ဖောင့်(ဖိုင်) မှတစ်ဆင့် woff, woff2, svg, eot ဖိုင် တွေကို ပြန်လည်ထုတ်ပေးလို့ရပါတယ်။ အဲဒါကို ခုကာက ကျွန်တော်တို့ Download လုပ်ထားတဲ့ Font နဲ့ သွားပြီး ထုတ်ကြည့်ရအောင်။ အခင်ဆုံး အောက်က လင့်ကို သွားကြပါမယ်။

<https://transfonter.org/>

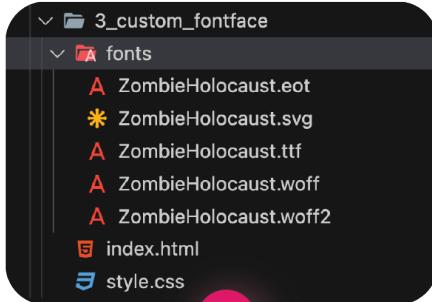


အညွှန်း (၁) ကိုစိပ်ပြီး ကျွန်တော်တို့ Download ခဲ့ထားတဲ့ ဖောင့်ကို ရွှေ့ပြီးတင်ပေးလိုက်ပါမယ်။ ပြီးရင် အညွှန်း (၂) အရ ပြန်ထုတ်မှာက ttf, woff, woff2, eot, svg အကုန်လုံးကို အမှန်ခြစ်ပြီး ထုတ်မှာပါ။ အညွှန်း (၃) အရ Convert လုပ်လိုက်ပြီဆိုတာနဲ့ ကျွန်တော်တို့ လိုချင်တဲ့ ဖိုင်တွေအကုန်လုံးက Generate လုပ်ပြီး အောက်ကပုံ အတိုင်းပေါ်လာမှာဖြစ်ပါတယ်။ Download ရွှေ့ချယ်လိုက်ပါ။





1



2

ဘယ်ဘက်က နံပါတ် (၁) ပုံကတော့ ကျွန်တော်တို့ Transfonter ကနေပြီး Download ဆဲပြီး ရလာတဲ့ ဖိုင်ပါ သူအထဲမှာ ZombieHolocaust ဖိုင်တွေကို woff2, woff, ttf, eot, svg ဆိုပြီး အကုန်လုံး Generate ထုတ်ပေးလိုက်တာကိုတွေ့ရမှာပါ။

ညာဘက်က နံပါတ် (၂) ပုံကတော့ ကျွန်တော်တို့ ပြုလုပ်ထားတဲ့ Folder ထဲကို နောက်ထပ် အသစ် Folder "fonts" ဆိုပြီး တည်ဆောက်လိုက်တာပါ။ ပြီးတော့ ခုက္ခဏက Generate လုပ်ထားတဲ့ fonts (၅) ခုကို ကူးယူလိုက်ပါမယ်။ ဒါဆိုရင် ကျွန်တော်တို့ နံပါတ် (၂) ပုံထဲကအတိုင်း အပြင်မှာ index.html , styles.css ရှုတဲ့ fonts ဆိုတဲ့ Folder ထဲမှာတော့ ခုက္ခဏက ကူးယူလိုက်တဲ့ Font Files (၅) ခုရယ်ရှိမှာပါ။

index.html အတွက် Gist Code ကို ကူးယူလိုက်ပါ။

<https://gist.github.com/lwinmoeapaing/0c074c299704601c35c961e3329533b6>

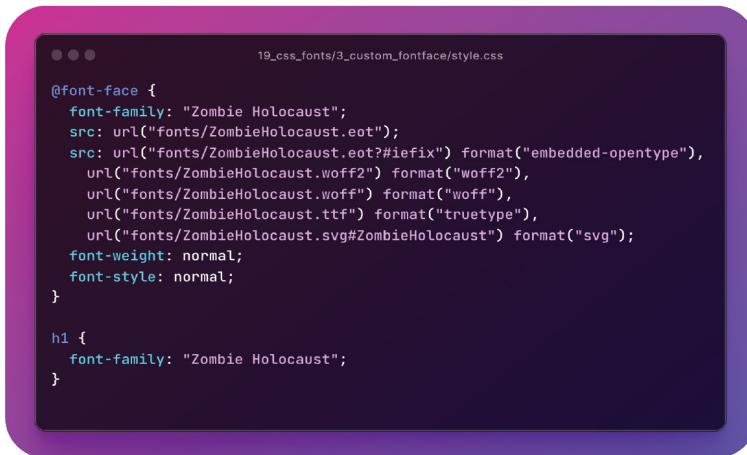
```
19_css_fonts/3_custom_fontface/index.html

... ● ● ●

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Custom Fonts | Lwin Moe Paing</title>
    <link rel="stylesheet" href="style.css" type="text/css" />
  </head>
  <body>
    <h1>Halloween Font</h1>
  </body>
</html>
```

style.css အတွက် Gist Code ကို ကူးယူလိုက်ပါ။

<https://gist.github.com/lwinmoepaing/8abd043d39818a2c2144602f4f2d9fc3>



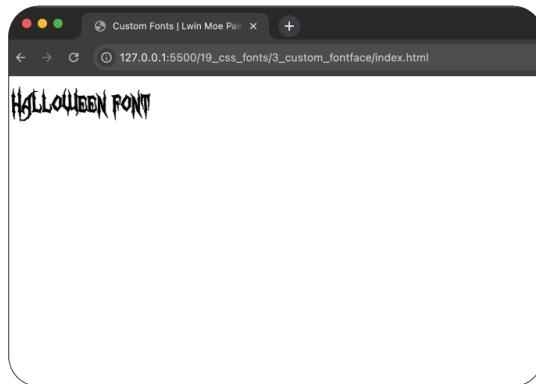
```

@font-face {
    font-family: "Zombie Holocaust";
    src: url("fonts/ZombieHolocaust.eot");
    src: url("fonts/ZombieHolocaust.eot?#iefix") format("embedded-opentype"),
        url("fonts/ZombieHolocaust.woff2") format("woff2"),
        url("fonts/ZombieHolocaust.woff") format("woff"),
        url("fonts/ZombieHolocaust.ttf") format("truetype"),
        url("fonts/ZombieHolocaust.svg#ZombieHolocaust") format("svg");
    font-weight: normal;
    font-style: normal;
}

h1 {
    font-family: "Zombie Holocaust";
}

```

စာဖတ်သူတို့ Browser ပေါ်မှာ Run လိုက်ပါက ဒီလိုပေါ်လာပါလိမ့်မယ်။



အခုခံရင် ကျွန်တော်နဲ့အတူတူ စာဖတ်သူတို့ကလည်း Halloween အတွက် Custom Font တစ်ခုထည့်သွင်းလိုပြီးသွားပါပြီ။ သို့ပေါ်မယ် ကျွန်တော်တို့ style.css ထဲမှာ ရေးသားထားသော Rule တွေကို တစ်ချက်လေ့လာကြည့်ရအောင်။

သူမှာ @font-face { ... } အရေးအသားနဲ့ ရေးထားတာကို သတိပြုပေးပါ။ ပထမတစ်ခါတုန်းကလည်း @import ဆိတ်ကို မြင်နှုပြီးသားဖြစ်မှာပါ။ "@" Symbol နဲ့စတဲ့အခါသူတို့က Selector မဟုတ်ကြောင်းအရင် သိရပါမယ်။ CSS မှာ တွေားသော Rule တွေကို အရင်ကြော်တဲ့အခါ "@" နဲ့စကြတယ်လို့သာ သိထားလိုက်ပါ။ ဆိုတော့ အထူး မှာ font-family အနေနဲ့ "Zombie Holocaust" ကို ကြော်လိုက်ပါတယ်။

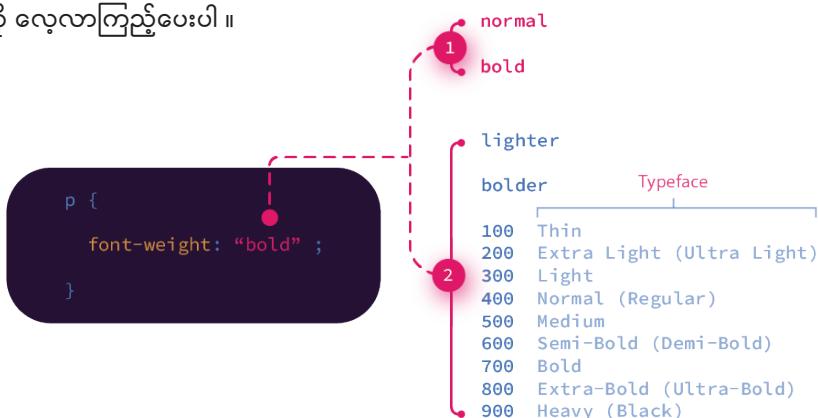
ပြီးတဲ့အခါမှာ src: url (...) ဆိုပြီး ကျွန်ုတ်တို့ ခုကာက Generate ထုတ်ထားတဲ့ ဖောင့် (၅) ခုရှိရာကို ထည့်သွင်းပြီး သတ်မှတ်ပါတယ်။ ဖိုင်ထည့်ရာမှာ Path (ဖိုင်ရှိရာ) လမ်းကြောင်းမမှားအောင် စာဖတ်သူတို့က သတိထားပေးပါ။ "fonts" folder ထဲ ကျွန်ုတ်တို့က ထည့်ထားတဲ့အတွက် အောင် "fonts/" ပုံစံနဲ့ url () ထဲမှာ Path လမ်းကြောင်းက စနေမှာဖြစ်ပါတယ်။ ပထမ src အတွက်တော့ eot ဖိုင်သာ ထည့်ထားခြင်းဖြစ်ပါတယ်။ ဒုတိယ src ကနေစပြီး url () နဲ့ format () ဆိုတဲ့ အရေးအသားနဲ့ ရေးသားပါတယ်။ ဒါတွေကို အလွတ်မှတ်ထားရမလားဆိုရင် စာရေးသူကိုယ်တိုင်လည်း အလွတ်မရပါဘူး။ Custom Font ထည့်မယ်ဆို ဒီလို့ထည့်သွင်းအသုံးပြု ရေးသားရတယ်ဆိုတာကို သိထားရင်ရပါပြီ။

Text Style များကို လေ့လာခြင်း

Font တွေဖြောင်းလဲပြီးနောက် CSS ကိုအသုံးပြုပြီး လက်ရှိ စာသားတွေ Text တွေကို Style Rule သတ်မှတ်ပြီး အသုံးပြုပုံကိုလေ့လာကြပါမယ်။

- Font Weight အသုံးပြုခြင်း

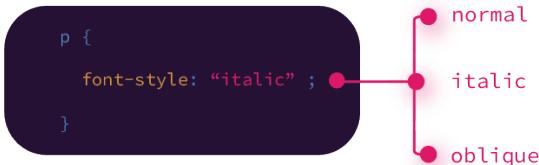
စာသားတွေရဲ့ Thickness နဲ့ Bold အထူး၊ အပါး ပြုလုပ်ချင်တဲ့အခါ အသုံးပြုပါတယ်။ ရေးသားပုံက Property "font-weight" ကို အသုံးပြုပြီး ရေးပါတယ်။ အောက်က ပုံလေး ကို လေ့လာကြည့်ပေးပါ။



အဓိကအားဖြင့်တော့ normal နဲ့ bold တန်ဖိုးပဲသုံးကြပါတယ်။ Typeface အခန်းမှာတုန်းက typeface နဲ့ font style ကိုရှင်းပြပြီးဖြစ်ပါတယ်။ ပုံထဲက အညွှန်း (၂) အတိုင်း lighter, bolder, 100 ~ 900 ကြား နံပါတ်တွေလည်းပေးနိုင်ပါတယ် သို့ပေါယ် Font ကိုယ်တိုင်က typeface font-style များကို Support မှ မဲ့ တန်ဖိုးတွေက အလုပ်လုပ်မှာဖြစ်ပါတယ်။ မဟုတ်ရင် ထူးချွာမှုရှိမှာမဟုတ်ပါဘူး။ "bold" တန်ဖိုး တစ်ခုသာ အလုပ်လုပ်သွားမှာပါ။

- Font Style အသုံးပြုခြင်း

တာသားတွေရဲ့စကာလုံးကို တာလုံးတောင်း Italic Style နဲ့ဖော်ပြချင်တဲ့အခါမျိုးမှာ အသုံးပြုပါတယ်။ ရေးသားပုံက Property “font-style” ကို အသုံးပြပြီး ရေးပါတယ်။ အောက်က ပုံလေးကို လေ့လာကြည့်ပေးပါ။



CSS property “font-style” အတွက် italic ကော် oblique ကော် တန်ဖိုး (J) ခုစာလုံးက စကားလုံးတောင်းပုံစံရအောင် ပြုလုပ်ပေးဖို့ သုံးနိုင်ပါတယ်။

ကွားခြားချက်ကတော့ italic ဆိုတဲ့ Rule က font creator (ဖော်ဖန်တီးပြုလုပ်သူက) version ကဲ့ variant အနေနဲ့ italic အတွက် သီးသန့်ဖောင့်ကို ပြုလုပ်ပေးထားတဲ့အခါ မျိုးမှာ အလုပ်လုပ်ပေးတာဖြစ်ပါတယ်။

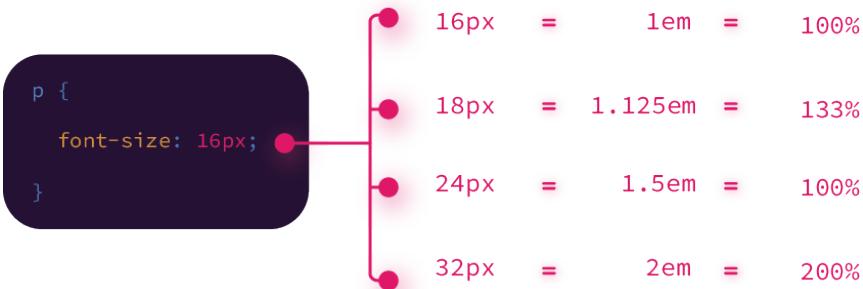
တစ်ချို့ Font တွေမှာ font creator (ဖော်ဖန်တီးပြုလုပ်သူက) သီးသန့် စကာလုံး တောင်းဖောင့်ကိုပြုလုပ်ပေးထားခြင်းမရှုပါဘူး။ ဒီလိုအခြေအနေမျိုးမှာ oblique ကို အသုံးပြုပြီး စကားလုံးတောင်းပုံစံ (italic appearance) ဖြစ်အောင်ပြုလုပ်နိုင်ပါတယ်။ သူကတော့ italic သီးသန့်ဖောင့်ကို လိုက်ရှာနေမှုမဟုတ်တော့ဘဲ မူရင်းလက်ရှိ ဖောင့်ကို အတောင်းဒီရိုင်းကိုပြောင်းပေးလိုက်တာဖြစ်ပါတယ်။ ပုံမှန်အားဖြင့်တော့ အတူတူ ပါပဲ။ Web လောကမှာ တွေ့ရများတဲ့ စကားလုံး အတောင်းတွေကတော့ oblique တွေ ဖြစ်ကြပါတယ်။

- Text တွေအတွက် Units များ

Default အနေနဲ့ Browser အပေါ်မှာ 16 pixel scale နဲ့သွားနေပါတယ်။ ဆိုလိုချင် တာက ကျွန်တော်တို့ စရေးလိုက်တဲ့ <rp> element ထဲက တာသားတွေက 16px ရှိတယ် လို့ဆိုလိုချင်တာပါ။ Property နာမည်ကို “font-size” နဲ့ အသုံးပြုပြီး၊ တန်ဖိုးအရ Unit ကို ဖောင့်တွေမှာ “px”, “em”, “%” ဆိုပြီး (၃) မျိုးကို အသုံးပြနိုင်ပါတယ်။

“em” ကို em-relative-length unit လို့ခေါ်ပါတယ်။

နောက်တစ်မျက်နှာမှာ ပုံနှင့်တကွ လေ့လာကြည့်ရအောင်...



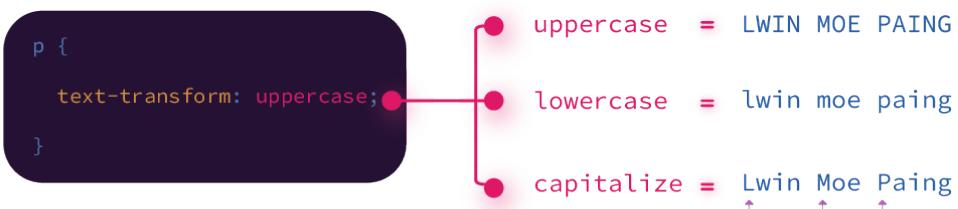
Equation အရဆိုလျင် Browser ရဲ့ Default Body Text က 16px ရှိပါတယ်။ ဒါကို 1em က 16px နဲ့ ညီမျှပါတယ် 100% ဖြင့်လည်းခေါ် သုံးလို့ရပါတယ်။ အောက် ဆုံး ဆက်သွယ်ချက် 32px ကို ကြည့်လိုက်ပါ 16px ရဲ့ နှစ်ဆဖြစ်သွားပါပြီ။ 16px က 1em ဆိုတော့ 32px ဆိုတော့ (၂) ဆဖြစ်သွားတာကြောင့် 2em နဲ့ ညီမျှပါတယ် 200% ပုံစံနဲ့လည်းအသုံးပြနိုင်ပါတယ်။

မှတ်ချက်။ 1em က အမြတ်စီမံခြေနေမှာတော့မဟုတ်ပါဘူး 100% ဖြစ်နေ မှာလည်းမဟုတ်ပါဘူး။ body { font-size: 16px } က Browser က သတ်မှတ်ပေးတာပါ။ စာဖတ်သူတို့က body { font-size: 12px } ပြောင်းလဲလိုက်ရင် 1em = 12px ဖြစ်သွားပါပြီ။ ဒါဆိုရင် 2em = 24px ဖြစ်သွားမှာပါ။ ကျွန်ုတ်တို့က လိုသလိုပြောင်းလဲနိုင်ပါတယ်။ သတ်ထားရမှာပါ သတ်မှတ်လိုက်တဲ့ font-size နဲ့ em ရယ် % ရယ်က တိုက်ရှိက ဆက်နှယ်မှုရှိနေပါတယ်။

- Text Transform စကားလုံးတွေကိုပြောင်းလဲမယ်

စကားလုံးတွေကို Uppercase (အကြီးစကာလုံး) ပြောင်းခြင်း (သို့မဟုတ်) အသေး စကားလုံးများ ပြောင်းချင်ပါက text-transform ကို အသုံးပြုပါတယ်။

● Example Text - “Lwin moe paing”



uppercase တန်ဖိုးကတော့ စကားလုံးတိုင်းကို အကြီးကြောင်းချင်တဲ့အခါ သုံးပါတယ် ॥ lowercase ကတော့ စကားလုံးတိုင်း အသေးကြောင်းချင်တဲ့အခါ အသုံးပြု နိုင်ပါတယ် ॥ capitalize ကျတော့ အရှေ့ဆုံး စကားလုံးများသာ အကြီး Capital Letter အဖြစ်မြင်ချင်တဲ့ အခါသုံးပါတယ် (ဥပမာ - အများအားဖြင့် နာမည်များကို ပြန်လည် ဖော်ပြရာတွင် အသုံးများပါတယ်) ॥

- Text Decoration စကားလုံးများ

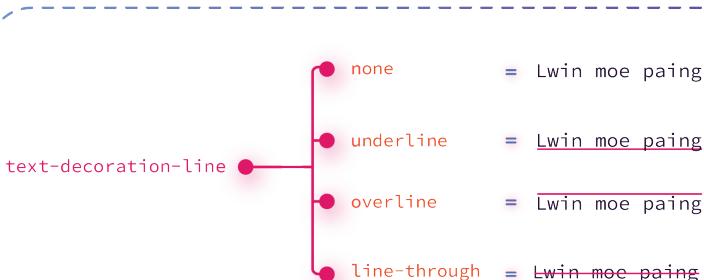
တသားများအား မျဉ်းလိုင်းများသုံးပြီး အောက်ခြေမှာ မျဉ်းတာတာမျိုး (သို့မဟုတ်) အလယ်ကနေ ကန့်လန်ဖြတ်တားခိုင်းများတွင် အသုံးပြုပါတယ် ॥ အောက်က ပုံကိုတစ်ချက်လေ့လာကြည့် ရအောင် ။

- Example Text - “Lwin moe paing”

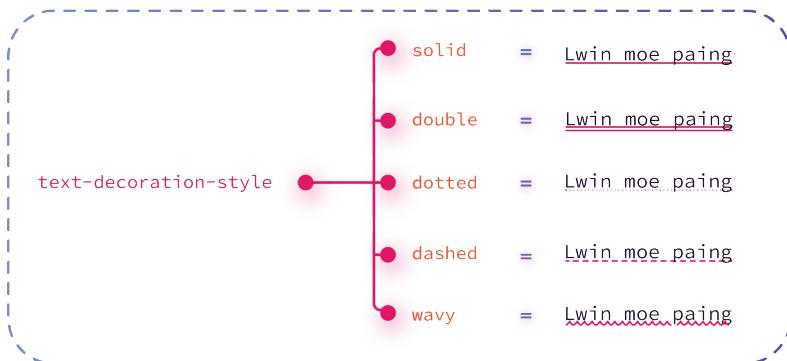
```
p {
    text-decoration-line: underline;
    text-decoration-color: red;
    text-decoration-style: solid;
    text-decoration-thickness: 2px;
}
```

text-decoration-line ဆိုတာက မျဉ်းကြောင်းလိုင်းရဲ့ တည်နေရာပါ ॥
 text-decoration-color ကတော့ မျဉ်းရဲ့အရောင်ကို ရွှေးချယ်ပေးရမှာဖြစ်ပြီး ।
 text-decoration-style ကတော့ မျဉ်းရဲ့ style ပါ solid လိုအတူလိုင်းလား (သို့မဟုတ်)
 dotted လိုင်းလိုမျိုး အစက်လေးတွေနဲ့ လိုင်းမျိုးလား ဆိုပြီး အသုံးပြုကြတာဖြစ်ပါတယ်
 text-decoration-thickness ကတော့ မျဉ်းလိုင်းရဲ့ အထူကို ရွှေးပေးရတာဖြစ်ပါတယ် ॥

အောက်ကပုံမှာ text-decoration-line မျဉ်းရဲ့ တည်နေရာကို လေ့လာကြည့်ရအောင်



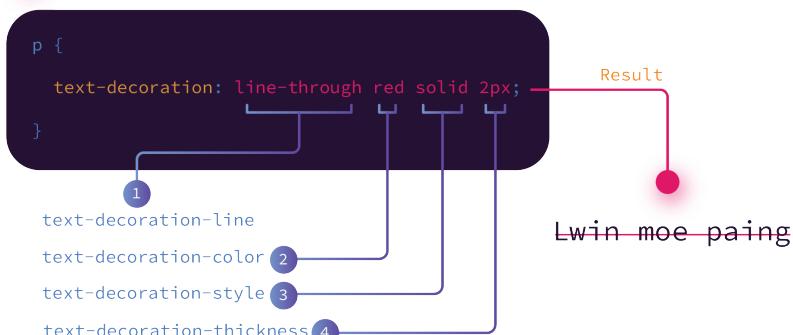
text-decoration-line မျဉ်းရဲ့ တည်နေရာမှာ none တန်ဖိုးသည် Default ပင်ကို တန်ဖိုးပါ မျဉ်းကြောင်းကို သတ်မှတ်မထားပါဘူး။ underline တန်ဖိုးက စာသားရဲ့ အောက်မှာ တားတဲ့အခါအသံးပြုပါတယ်။ overline ကတော့ စာသားရဲ့ အပေါ်မှာ မျဉ်းကြောင်းကို အသံးပြုချင်တဲ့အခါ ရွေးချယ်နိုင်ပါတယ်။ line-through ကျတော့ စာသားရဲ့ အလယ် တည့်တည့်ကနေ ဖြတ်ချလိုက်တာပါ။ (အမှား စာသားများ သို့မဟုတ် အကြော်ကို ပြချင်တဲ့အခါမျိုးမှာ အသံးများပါတယ်)



text-decoration-line ကို မျဉ်းရဲ့ style ကိုရွေးချယ်ချင်တဲ့အခါ အသံးပြုပါတယ်။ solid က အထူလိုင်း၊ double က (J) ဆ ဖြစ်တဲ့အတွက် လိုင်း (J) ကြောင်း၊ dotted အစက်လိုင်း၊ dashed ကတော့ “_” လိုင်းမျိုးပါ၊ wavy က လိုင်းတွန်းလေးတွေနဲ့ မျဉ်းလိုင်းမျိုးဖြစ်ပါတယ်။

text-decoration ကို အသံးပြုရာမှာ အတိကောက်အရေးသားအနေနဲ့ အောက်က ပုံ အတိုင်း ရေးသားနိုင်ပါတယ်။

● Example Text - “Lwin moe paing”



(၁) က လိုင်းနေရာ၊ (၂) က အရောင်၊ (၃) က style ဖြစ်ပြီး၊ (၄) ကတော့ မျဉ်းရဲ့အထူပါ အားလုံးကို Space “ ” ပြားပြီး ရေးသားရမှာပါ။

- Text Line-Height ကိုလေ့လာခြင်း

line-height စာကြောင်းတစ်ကြောင်းနဲ့ တစ်ကြောင်းကြား အကွာအဝေးသတ်မှတ်ရာတွင် အသုံးပြုပါသည်။ Browser တစ်ခုရဲ့ Default ပုံမှန်သတ်မှတ်ထားတာက 110% ~ 120% ကြားဖြစ်ပါတယ် ဖြစ်ပါတယ်။ အောက်က ပုံကို ကြည့်ကြည့်ပါ line-height : 1 ရဲ့ အဓိပ္ပာယ်က 100% နဲ့ အတူတူပဲဖြစ်ပြီး လက်ရှိ ဖောင့်ဆိုင် ရဲ့ တစ်(ဆ) လို့ဆိုလိုတာပါ။ ပထမပုံကို ကြည့်ပါ စားကြောင်း တစ်ကြောင်းနဲ့ တစ်ကြောင်း အကွာအဝေးက ကပ်နေပါတယ်။ ဒုတိယပုံမှာတော့ ကြားထဲအကွာအဝေးကို 1.3 (130%) လုပ်လိုက်တဲ့ အပါ (30%) နှုန်းက ကြားထဲမှာ နေရာယူသွားမှာပါ။

<p>

 Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi.

</p>

line-height: 1

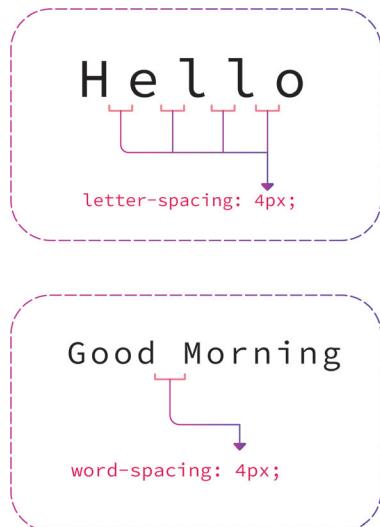
<p>

 Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat. Duis autem vel eum iriure dolor in hendrerit in vulputate velit esse molestie consequat, vel illum dolore eu feugiat nulla facilisis at vero eros et accumsan et iusto odio dignissim qui blandit praesent luptatum zzril delenit augue duis dolore te feugait nulla facilisi.

</p>

line-height: 1.3

- Text Letter Spacing နဲ့ Word Spacing



letter-spacing က စကုလုံး တစ်လုံးနဲ့ တစ်လုံးကြားထ အကွာအဝေးကို ပြားချင်တဲ့ အခါအသုံးပြုပါတယ်။ word-spacing ကတေသာ စကားရ တစ်စုနဲ့ တစ်စုကြား (Space Bar) ပြားထားတဲ့ စကားစု၏ အကွာအဝေးများကို ပြားချင်တဲ့ အခါအသုံးပြုပါတယ်။

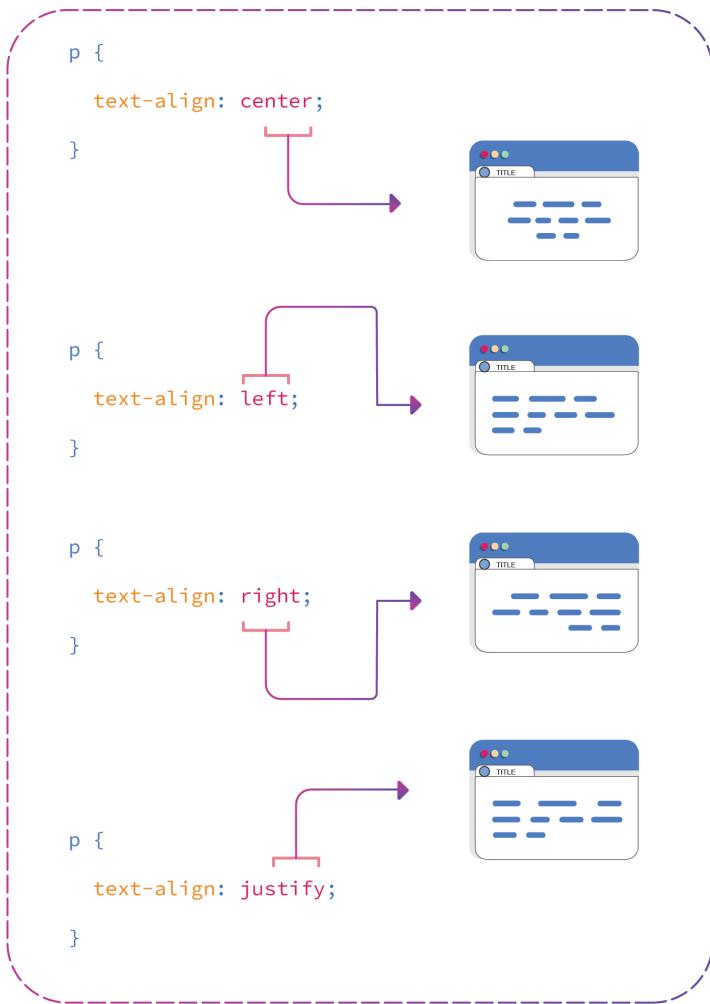
- Text တွေကို Alignment ခိုန်ခြင်း

Property (text-align) ကို အသုံးပြုပြီး စာသားများရဲ့ တည်နေရာကို ပြောင်းလဲ သတ်မှတ်နိုင်ပါတယ်။ တန်းဖိုးတွေကို အောက်က အတိုင်း

1. left (Default တန်းဖိုးပါ ဘယ်ဘက်ကပ်နေစေပါတယ်)
2. right (ညာဘက် ကပ်နေစေပါတယ်)
3. center (စာသားများအလယ်မှာပေါ်ခေါ်ချင်တဲ့အပါ သုံးပါတယ်)
4. justify (နောက်တစ်လိုင်းဆင်းခါနီးမှာ ကြားထဲက လိုအပ်တယ့် Space တွေကိုဖြည့်ပေးသွားတဲ့ ပုံစံပါ)

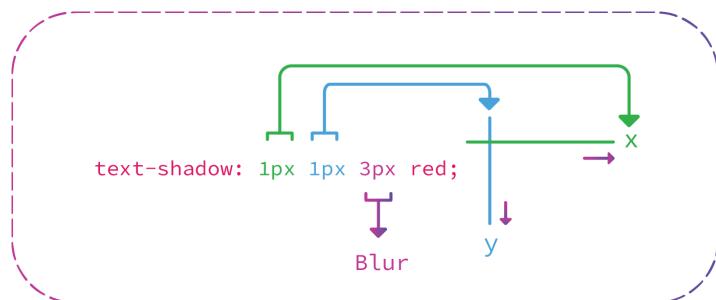
တန်းဖိုး (၄) မျိုးကို တွေ့ရမှာပါ။

Property ကို အသုံးပြုရောတွင် selector { text-align: center } ပုံစံဖြင့် သုံးလေ့ရှိပါတယ်။
အောက်ကပ်လေးကို တစ်ချက်လေ့လာကြည့်ပေးပါ။

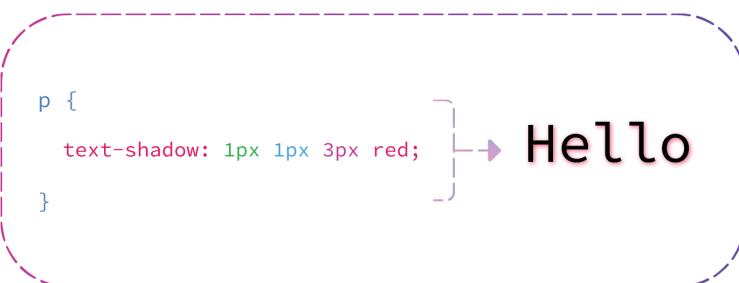


- Text များအတွက် အရိပ်

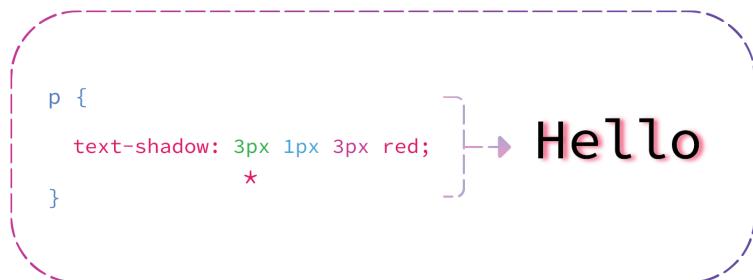
စာသားတွေရဲ့အနောက်မှာ Shadow အရိပ်ကို ဖော်ပြချင်တဲ့အခါမှာ text-shadow ဆိုတဲ့ Property ကိုအသုံးပြုပြီးရေးသားနိုင်ပါတယ်။ text-shadow ကို တန်ဖိုးပေးတဲ့အခါမှာ X-offset တန်ဖိုး၊ Y-offset တန်ဖိုး၊ Blur Radius (မှုန့်ဝါးစေသော) တန်ဖိုးနဲ့ Color (အရိပ်အရောင်) တန်ဖိုးတို့ကို ထည့်ပေးရမှာပါ။ အောက်က ရေးသားနည်းကို လေ့လာကြည့်ပါ။



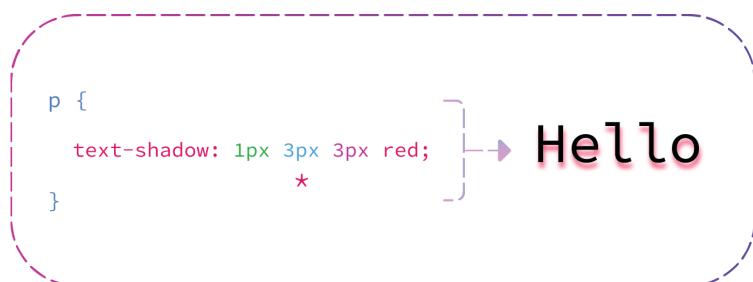
ပထမ တန်ဖိုးက Horizontal လုံးမှာရှိတဲ့ X တန်ဖိုးပါ သူကများလေ အရိပ်က ညာဘက်ကိုရောက်သွားလေဖြစ်မှာပါ။ ဘယ်ဘက်ကို တိုးချင်ရင် (-) minus တန်ဖိုးသုံးပြီး ပြုလုပ်လို့ရပါတယ်။ ဒုတိယ တန်ဖိုးက Y တန်ဖိုးပါ တန်ဖိုးများလေ အောက်ကို ရောက်သွားလေဖြစ်မှာပါ။ တတိယ တန်ဖိုးကတော့ Blur Radius တန်ဖိုးပါ။ နောက်ဆုံးတစ်ခုကတော့ အရောင်ရဲ့ တန်ဖိုးကို ထည့်ပေးရမှာဖြစ်ပါတယ်။ အောက်က ပုံကိုကြည့်ပြီး လေ့လာပေးပါ။



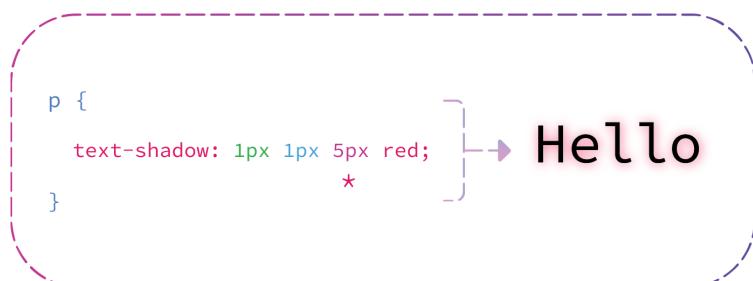
Hello ဆိုတဲ့ စာသားရဲ့ အနောက်မှာ အနီရောင် အရိပ်ကို ပေါ်ရန် အသုံးပြုထားပုံဖြစ်ပါတယ်။ နောက်တစ်မျက်နှာမှာ X တန်ဖိုးကို တိုးပြီး အသုံးပြုထားပုံကို လေ့လာကြည့်ရအောင် ...။



X တန်ဖိုးကို 1px မှ 3px တိုးပြီး အသုံးပြုထားလိုက်တဲ့ အခါ အရိပ်က ညာဘက်ကို တိုးသွားမှာဖြစ်ပါတယ်။



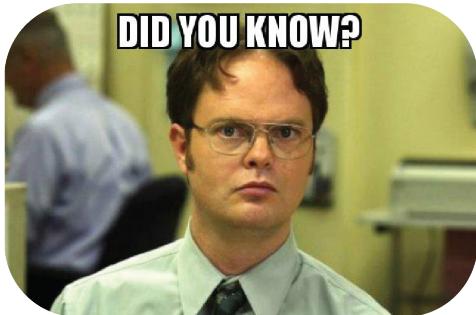
Y တန်ဖိုးကို 1px မှ 3px တိုးပြီး အသုံးပြုထားလိုက်တဲ့ အခါ အရိပ်က အောက်ခြေဘက် ကို တိုးသွားမှာဖြစ်ပါတယ်။



Blur Radius တန်ဖိုးကို တိုးလိုက်တဲ့ အခါမှာတော့ အနောက်က အရိပ်ရဲ့ (မျှန်ဝါးစက်ဝန်း) အကျယ်က တိုးသွားတာကြောင့် Hello စာသားက ပိုပြီး အရိပ် အရောင်မှာ ဝါးသွားမှပါ။

CSS ၄။ Font တွေအကြောင်း Text တွေအကြောင်းကို ကျွန်တော်တို့ လေ့လာသင့် သလောက် လေ့လာပြီးသွားပြီးဖြစ်တဲ့အတွက် စာဖတ်သူများကို Congratulation ပါလို့ ပြောချင်ပါတယ်။

နောက်တစန်းမှာ CSS ၅။ Other Selectors တွေအကြောင်း လေ့လာသွားကျေရမှာပါ။



သင်သိပါသလား ?

စာရေးသူက ပထမတုန်းကလည်း CSS Selector တွေကို ရှင်းပြပြီးသွားပြီဆိုတာ ။ အခုလည်း နောက်ထပ်စာဖတ်သူတွေကို ထပ်မှတ်ခိုင်းပြီးမှာပါ ။

Attribute Selector များနှင့် Pseudo Class

Pseudo-Classes များအကြောင်း

Pseudo-Class တွေကို Element တစ်ခုရဲ အခြေအနေအမျိုးမျိုးနဲ့ ပတ်သက်ပြီး အသုံးပြုကြပါတယ်။ Element ရဲ၊ Specific State တစ်ခုခုတိကျတဲ့ အခြေအနေမျိုးမှာ လုပ်ဆောင်စေချင်တာမျိုး Style Rule သတ်မှတ်ပေးချင်တာမျိုးမှာ သုံးပါတယ်။ ဥပမာ - Button ခုတုတ်လေးကို Mouse နဲ့ တင်လိုက်တဲ့အခါ ခလုတ် အရောင်ပြောင်း သွားတာမျိုး အဲလိုအခြေအနေမှာသုံးတဲ့ Class ပါ။ နောက်တစ်ခုက Condition Element ကို မှတ်ညွှန်ပြီးတော့ လည်း သုံးတတ်ကြပါတယ် ဥပမာ - Table Element <table> ထဲမှာ <tr> Table Row တွေရှိပါတယ် အဲ Row တွေထဲမှာ စုံကေန်းပိုင်ဆိုင်တဲ့ Row တွေရဲ့ နောက်ခံအရောင်ကိုပေးချင်တဲ့တော် သုံးခြင်းဖြစ်ပါတယ်။ ဆိုလိုချင်တာက တစ်လိုင်းကော်တိုင်း အရောင်ပြောင်းချင်တာမျိုးဆိုရင် Pseudo-Class တွေ အသုံးပြုလိုရပါတယ်။

Pseudo Class တွေက များပါတယ် တကယ့်လက်တွေ့မှာ အသုံးပြုအများဆုံးတွေကို ပဲ ခိုစာအပ်မှာဖော်ပြပေးသွားမှာပါ။ ပြင်ပ လုပ်ငန်းခွင်မှာလည်း တရေးသူကိုယ်တိုင် အကုန်လုံးမမှတ်မိဘလို လိုအပ်လာလျှင် Documentation တွေသွားဖတ်ရပါတယ်။ စာဖတ်သူတွေလည်း အကုန်မှတ်မိနိုင်မှာမဟုတ်ပါဘူး လေ့လာရင်း လိုအပ်လျှင် သွားရောက်လေ့လာပြီး ရေးသားကြရမှာပါ။

- Hyperlink <a> အတွက် Pseudo-Class များ

<a> Element ကို Link တွေကို တစ်ခုနဲ့ တစ်ခုချိတ်ဆက်ဖို့အတွက် အသုံးပြုနိုင်တယ် ဆိုတာကို HTML အခန်းမှာ လေ့လာပြီးသွားပါပြီ။ Browser မှာ ပုံမှန်အားဖြင့် Element <a> ကိုအသုံးပြုရော့မှာ underline ဖြစ်တဲ့ အောက်မှာ မျဉ်းကြောင်းတစ်ကြောင်းရယ် စာသားကို အပြောရောင်အနေနဲ့ တွေ့ရမှာဖြစ်ပါတယ်။ အကယ် စာဖတ်သူတိုက Link ကို တခါနိုင်ပြီး ဝင်သွားတဲ့အောက် လက်ရှိ Page ကိုပြန်လာခဲ့ရင် အပြောရောင်မဟုတ် တော့ဘဲ ခရမ်းရောင်ဖြစ်သွားမှာပါ။ Visited အသုံးပြုသွား Link ကိုတစ်ခါ သွားပြီးဖြစ်တဲ့အခါ ခရမ်းရောင်ဖြစ်သွားခြင်းဖြစ်ပါတယ်။ အဲလိုမျိုးတွေကို Pseudo-Class သုံးပြီး CSS Rule တွေသတ်မှတ်ပေးနိုင်ပါတယ်။

www.google.com
အသုံးပြုသူ Link ကိုမှန်ပိုင်



www.google.com
အသုံးပြုသူ Link ကို နှိပ်ပြီး

User Experience (UX) အတွက်ကောင်းပြီး အသုံးပြုသွား ခရမ်းရောင်မြင်တာနဲ့ ဒီလုံးကို သွားပြီးပြီဆိုတာသိမှာပါ။

ပထမဆုံးအနေနဲ့ Pseudo-Class တစ်ခုကိုရေးရာမှာ ":" ကော်လု (Full Column) ကို
ထည့်သွင်းရေးသားလေ့ရှိပါတယ်။

အခုပထမအနေနဲ့ ":link" ကို ချိတ်မဆက်ခင် အသုံးပြုသူက မနိုင်ခင် အနေအထား
မှာသုံးနိုင်တဲ့ Pseudo Class ကို လေ့လာကြည့်ပါမယ်။ ပြီးရင် နိုင်ပြီးသွားတဲ့
အနေအထားမှာ သုံးတဲ့ ":visited" ကိုလေ့လာပါမယ်။

```
<a href="/"> Go To Link </a>
```

အရင်ဆုံး HTML ဖိုင်ထဲမှာ <a> element လေးတစ်ခုတည်ဆောက်လိုက်ပါမယ် ပြီးရင်
အောက်က ပုံတွေအတိုင်း CSS Rule ကိုပေးလိုက်ပါမယ်။

```
a:link {  
  
    color: white;  
    background-color: #de1767;  
    padding: 10px;  
    border-radius: 8px;  
  
}
```

Go to Link

ဒီပုံထဲမှာဆိုရင် Go To Link ဆိုတဲ့ <a> Element ကို အသုံးပြုသူက မနိုင်ရသေးခင်ပုံ
ဖြစ်ပါတယ်။ သူရဲ့ ရေးသားနည်းကို ကြည့်ကြည့်ပါ "a:link { }" ပုံစံပါ။ အလယ်မှာ ":"
Full Column နဲ့ အနောက်မှာက်ပိုပို "link" ရှိပါတယ်။ ခွာပြီးရေးသားလိုမရပါဘူး။
{ } အထူးအစားတွေကို နောက်ခံ အရောင်၊ စာသားအရောင်များ ရေးထား
တာကိုတွေ့ရမှာပါ။ အောက်ကပုံကို ထပ်လေ့လာကြည့်ရအောင် ...

```
a:visited {  
  
    color: white;  
    background-color: #241233;  
    padding: 10px;  
    border-radius: 8px;  
  
}
```

Go to Link

ဒါကတော့ အသုံးပြုသူက နိုင်လိုက်တာနဲ့ နောက်ခံအရောင် (Background Color) ကို
နက်ပြာရောင်အဖြစ်ထားမယ်လို့ ရေးထားတာဖြစ်ပါတယ်။ အသုံးပြုသူက အဲ လိုက်ပါဘူး။
သွားရောက်လည်ပတ်ပြီး (နိုင်ပြီး) ချိန်တွင် ထို CSS Rule က အလုပ်လုပ်သွားမှာပါ။

- Action အကြခါန်ကိုဖမ်းသော Pseudo-Class များ

:hover, :active, :focus အစရိတ္တဲ့ Pseudo-Class တွေဖြစ်ပါတယ်။

:hover ကို စပြီးလေ့လာကြည့်ပါမယ်။ :hover က ဘယ်အကြခါန်မှာ အလုပ်လုပ်သလဲဆိုရင် User(အသုံးပြုသူ) က Mouse Cursor မောက်လိုင့်တာ မြှေးလေးနဲ့ Selector လုပ်ထားတဲ့ Element အပေါ် တင်လိုက်တဲ့အခါ (သို့) ရောက်သွားတဲ့ အခါ အလုပ်လုပ်ပါတယ်။

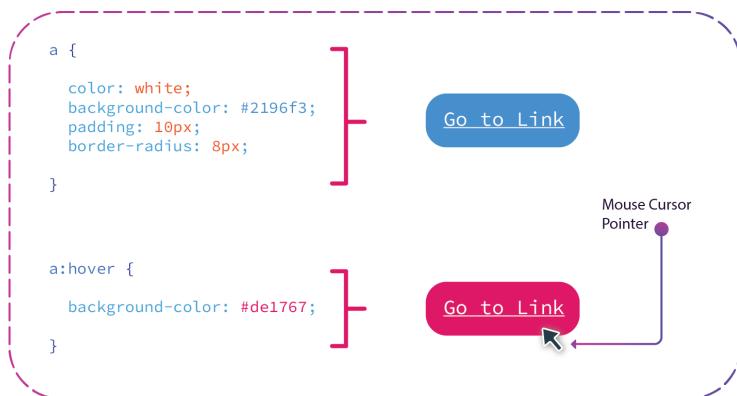
ဥပမာ - အဗြာရောင် Button ခလုတ်လေးတစ်ခု ရှိတယ်ဆိုပါစို့။

အသုံးပြုသူက Click မနိုင်ခင် Mouse Cursor (မြှေး) အပေါ်တင်လိုက်တဲ့အချင့်မှာ ခရမ်းရောင် Button ခလုတ် အဖြစ် အရောင်ပြောင်းချင်တာမျိုးမှာသုံးပါတယ်။

မှတ်ချက်။ မှန်-စခရင် (Screen) များတွင် Mouse နဲ့အသုံးမပြုနိုင်သောကြောင့် Mouse Pointer မပါရှိခြင်းကို သတိပြုပေးပါ။

```
<a href="/"> Go To Link </a>
```

အရင်ဆုံး HTML ဖိုင်ထဲမှာ <a> element လေးတစ်ခုတည်ဆောက်လိုက်ပါမယ် ပြီးရင်အောက်က ပုံတွေအတိုင်း CSS Rule ကိုပေးလိုက်ပါမယ်။



ပထမအနေနဲ့ <a> Element ကိုပဲ အဗြာရောင် ခလုတ်လေးအဖြစ် ကျွန်ုတ်တို့ CSS Rule တွေသတ်မှတ်ပေးလိုက်ပါတယ်။ နောက်ထပ် :hover ကတော့ Mouse Cusor ဖြစ်တဲ့ Pointer မြှေးလေး ခလုတ် အပေါ်ကိုတင်လိုက်တဲ့အချင့်မှာ နောက်ခံအရောင်အနီးရောင်အဖြစ်ပြောင်းပေးသွားမှာပါ။

:active ကျတော့ Element တစ်ခုကို နိပ်လိုက်တဲ့အချင့် (သို့) ဖုန်းစခရင်က နေ့ Element ကို လက်နဲ့ မိလိုက်တဲ့အချင့်မျိုးမှာ အလုပ်လုပ်ပါတယ်။ ဥပမာ - အပြောရောင် Button စလုတ်တစ်ခုကို User အသုံးပြုသူက Mouse နဲ့ Click လိုက်တဲ့အခါ (သို့မဟုတ်) မိတားလိုက်တဲ့အခါ အပြောရောင်ကိုနည်းနည်းမိန့်ပေးလိုက် တယ် အကယ်သူက Click (သို့မဟုတ်) မိတားခြင်းကို ပြန်လွှတ်လိုက်တဲ့အခါ အရင် နဲ့ အပြောရောင်ကိုပြန်ပြောင်းသွားမှာပါ။ User အသုံးပြုသူကို ပိုပြီးတော့ နိပ်နေတဲ့နဲ့ အသုံးပြုနေဆဲ ခံစားချက်ကို ရောချင်တဲ့အခါ အသုံးပြုပါတယ်။

```
<button type="button"> Button </button>
```

အရင်ဆုံး HTML ဖိုင်ထဲမှာ <button> element လေးတစ်ခုတည်ဆောက်လိုက်ပါမယ် ပြီးရင်အောက်က ပုံတွေအတိုင်း CSS Rule ကိုပေးလိုက်ပါမယ်။

```
button {
    cursor: pointer;
    padding: 10px;
    background-color: #fdfdfd;
    border-radius: 1px solid #dfdfdf;
    border-radius: 8px;
}

button:active {
    color: white;
    background-color: #241233;
}
```

by Lwin Moe Paing

Button ကို မိတားကြည့်လိုက်တဲ့အခါ အောက်ကဗုံ အတိုင်း အရောင်ပြောင်းသွားပည်

မိတားခြင်းကို ပြန်လွှတ်လိုက်တဲ့အခါ မူလ နဲ့အရောင်ကိုပြောင်းသွားပါမည်

Button

:focus က Input Element တစ်ခုကို Focus (အာရုံစိုက်ဆိုတာထက် ဒီနေရာမှာ အသုံးပြုနေအခါ) အသုံးပြုပါတယ်။ စာဖတ်သူတို့ HTML Element များဆိုတဲ့ အခန်းမှာ Form Control တွေကို လေ့လာခဲ့ပြီးဖြစ်ပါတယ်။

:focus Pseudo-Class ကို <input> , <select>, <textarea> အစရှိတဲ့ Form Elements များမှာ အသုံးပြုနိုင်ပါတယ်။

ဥပမာ - အသုံးပြုသူ နာမည် ရိုက်ထည့်ဖို့ Input Box လေးတစ်ခုရှိတယ် ဆိုပါစို့။ အသုံးပြုသူ User က Input Box ကို Click လိုက်ရင် စာရိုက်လို့ရသွားမှာဖြစ်ပါတယ် အဲအချိန်ကို Focus ဖြစ်သွားတယ်လို့ အလွယ်သတ်မှတ်လို့ရပါတယ်။ Input Box ကို အသုံးပြုနေတဲ့အချိန် (သို့) စာရိုက်ဖို့အတွက် ready အသင့်အနေအထားမှာရှုနေပြီ Focus Mode On သွားတဲ့အချိန်ပေါ့ အဲအခြေအနေမျိုးမှာ CSS Rule တွေကို သတ်မှတ်ပေးလို့ရပါတယ်။

```
<input class="input" type="text" name="name" />
```

အရင်ဆုံး HTML စိုင်ထဲမှာ <input> element လေးတစ်ခုတည်ဆောက်လိုက်ပါမယ် class="input" ကိုသတိထားပေးပါ Class Selector ဖြစ်တဲ့အတွက် CSS ရေးသားနည်းမှာ “.” သုံးပြီးရေးရမှာဖြစ်ပါတယ်။

နောက်တစ်မျက်နှာမှာ :focus အကြောင်း ပုံနှင့်တက္က လေ့လာကြည့်ရအောင် ...

```
.input {
    padding: 10px;
    background-color: #fdfdfd;
    border-radius: 1px solid #cbcbcb;
    border-radius: 8px;
}
```

Enter Name

```
.input:focus {
    outline: 2px solid #9217de;
}
```

Enter Name

Input ကို Click လိုက်တဲ့အခါ Focus Mode ဖြစ်သွားသဖြင့်
အောက်ကပု အတိုင်း outline ခရိုင်းရောင်ပေါ်လာမည်။
စာရိုက်နှင့်အတွက် Cursor လေးလည်းပေါ်သွားမည်။

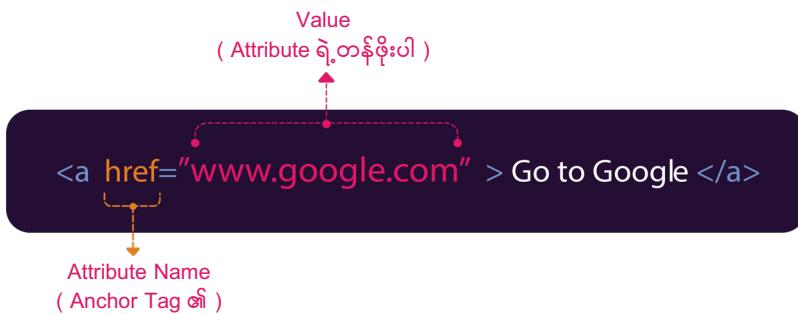
by Lwin Moe Paing

User အသုံးပြုသူက စာရိုက်ပြီးသွားသောလည်းကောင်း အခြား
Element (ဘုံး) Browser ရဲ့ တစ်နေရာရာကို Click လိုက်တဲ့အခါ
နိုင် မူရင်း Input Style ကိုပြန်ပြောင်းလည်းသွားပါမည်။

မှတ်ချက်။ ။ <input> နှင့် <textarea> Elements များသည် စာရိုက်ထည့်သည့် Element များဖြစ်သောကြောင့် Click လိုက်သောအခါ Cursor ပေါ်သွားပြီး Focus Mode ဖြစ်သွားတယ်လို့ လေ့လာသိရှိပြီးဖြစ်သည်။ သို့ပေမယ <select> Element သည် ရွေးချယ်ရတဲ့ Form Control Element ဖြစ်သောကြောင့် Click နှုပ်လိုက်သောအခါ အခြား Option Box များကျလာချိန်တွင် Focus Mode ဖြစ်သွားပါသည်။ စာဖတ်သူတို့ က class="input" တိုအသုံးပြုပြီး <select> Element နှင့် ကိုယ်တိုင် စမ်းသပ်နိုင်ပါသည်။

Attribute Selector များအကြောင်း

စာဖတ်သူတို့ ပြန်လည်မှတ်မိသွားအောက် အရင်ဆုံး အောက်က HTML Element လေးကို လေ့လာကြည့်ရအောင် Element တစ်ခုမှာ အဖွင့် Tag နဲ့ အပိတ် Tag ပါပြီး တော့ သက်ဆိုင်သာ Attribute တွေပါတယ်ဆိုတာကို သိထားပြီးသားဖြစ်ပါတယ်။ အခါ CSS နဲ့ သက်ဆိုင်ရာ Attribute တွေကို Select မှတ်ပြီး Rule တွေရေးကာ Style တွေ သတ်မှတ်လို့ရပါတယ်။



ဒီပုံလေးမှာဆိုရင် href က Attribute Name ဖြစ်ပြီး “www.google.com” ကတော့ Attribute Value ဖြစ်ပါတယ်။

ဒါဆိုရင် Attribute တွေကို Select လုပ်ပြီး ရေးသားတဲ့ နည်းလမ်းများကို နောက် တစ်မျက်နှာမှာ ကယားနှင့်တကွ လေ့လာကြည့်ရအောင် ...။

Attribute Selector များကို ဒီစာအုပ်မှာတော့ ကျွန်ုတ် Code တွေနဲ့ ရှင်းပြမှာမဟုတ်တော့ပါဘူး စာဖတ်သူတို့က လိုက်လုပ်ချင်ရင်တော့ ကိုယ်တိုင် စမ်းသပ် ရေးသားနိုင်ပါတယ်။

ရွေးချယ်မှု - Selector

အဓိပါယ်

ဥပမာ

EXISTENCE

[]

Attribute Name ရှိသော

Element များကို Select ဖုန်း။

a[target]

<a> Element တွေထဲမှာ Attribute name "target" ပိုင်ဆိုင်သော Element များ သက်ဆိုင်ပါတယ်။

eg - ...

EQUALITY

[=]

တိကျသော Attribute Name

ရယ် Attribute Value ရယ် ရှိ

သော Element များကို Select လုပ်ဖြင့်။

input[type="text"]

<input> Element တွေထဲမှာ Attribute name "type" ဘူး၏ Attribute value "text" ဖြစ်သော Element များနှင့်သာ သက်ဆိုင်ပါတယ်။

eg -

<input type="text" name="name" />

SPACE

[~=]

ကြေားလိုက Space များကိုပါ

ခွင့်ပြုပြီး Attribute Name ရယ်

Attribute Value ရယ်ရှိသော

Element များကို Select လုပ် ပြင်း

li[class~="red"]

 Element ရှိမှု Attribute name "class" ဘူး၏ Attribute value "red" ဖြစ်သော Element များနှင့်သာ သက်ဆိုင်ပါတယ်။ တိကျဝရာမလိုပဲ အလယ်တွင် Space များပါ နေလည်းရပါတယ်။

eg -

<li class="red">Apple

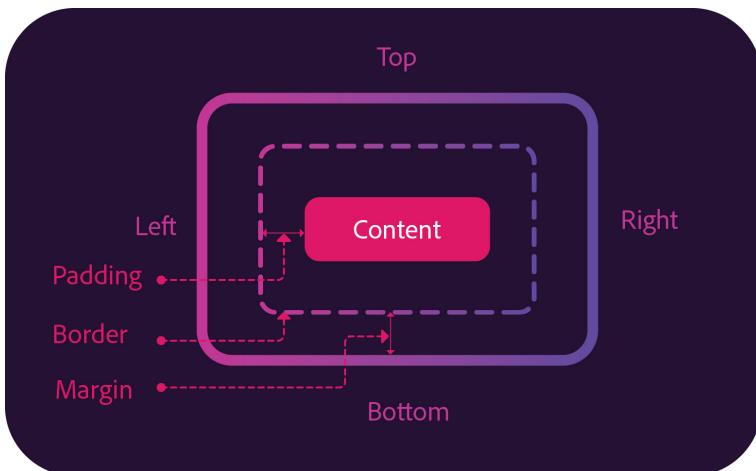
<li class="green red">Watermelon

<li class="red blue">Raspberry

class တိုင်းမှာ red ပါသောကြောင့်

li အားလုံးကို သက်ရောက်ပါမယ်။

| ရွှေးချယ်မှု - Selector | အဓိပါယ် | ပြမာ |
|-------------------------|--|---|
| PREFIX | [^=] Select Attribute Name အတွက် သတ်မှတ်လိုက်သော Value စာသားဖြင့် စေသာ Element များကို Select လုပ်ခြင်း။ | [class^="top"] Element တွေထဲမှာ Attribute name "class" ပိုင်ဆိုင်ပြီး Attribute Value သည် "top" စာသားဖြင့် အစပြောသော Element များအားလုံး သက်ဆိုင်ပါတယ်။ eg - <div class="top-left-border"> .. </div> <div class="top-right-border"> .. </div> <div class="topper"> .. </div> |
| SUBSTRING | [*=] Select Attribute Name အတွက် သတ်မှတ်လိုက်သော Value စာသား ပါသော Element များ ကို Select လုပ်ခြင်း။ | [class*="top"] Element တွေထဲမှာ Attribute name "class" ပိုင်ဆိုင်ပြီး Attribute Value သည် "top" စာသားပါဝင်သော Element များ အားလုံး သက်ဆိုင်ပါတယ်။ eg - <div class="top-left-border"> .. </div> <div class="stopper"> .. </div> <div class="bottom-top"> .. </div> class ထဲတွင် top စာသားပါသည့်နှင့် သက်ရောက်မှုပြုပါတယ် |
| SUFFIX | [\$=] Prefix နဲ့တောင်းပြန် Suffix က Select Attribute Name အတွက် သတ်မှတ်လိုက်သော Value စာသားဖြင့် အဆုံးသတ်သော Element များကို Select လုပ် ခြင်း။ကို Select လုပ်ခြင်း။ | [class\$="top"] Element တွေထဲမှာ Attribute name "class" ပိုင်ဆိုင်ပြီး Attribute Value သည် "top" စာသားဖြင့် အဆုံးသတ်သော Element များအားလုံး သက်ဆိုင်ပါတယ်။ eg - <div class="border-top"> .. </div> <div class="full_stop"> .. </div> <div class="bustop"> .. </div> |



HTML Element တိုင်းသည်
သူရဲ့ ကိုယ်ပိုင် Box လေးတွေထဲမှာနေတိုင်ပါတယ်။

Border ပေးလိုက်ရင် Border Line လေးဖြင့်ရမှာပါ။
Border အတွင်းသက်ကို အကွာအဝေး ခြားချင်တဲ့အခါ Padding သုံး
ပြီး။ အပြင်သက်ကို ခြားချင်တဲ့အခါ Margin သုံးပါတယ်။
Position အေရ Top, Left, Right, Bottom နဲ့လည်းရွှေ့လို့ရပါသေးတယ်
။ CSS နှင့် Boxes များအကြောင်းကို လေ့လာကြပါတော့မယ်။

သို့ပေမယ့် အခြားသူရဲ့ နှလုံးသား Box ကိုတော့ သွားမကစားကြပို့
သတိပေးလိုပါတယ်။

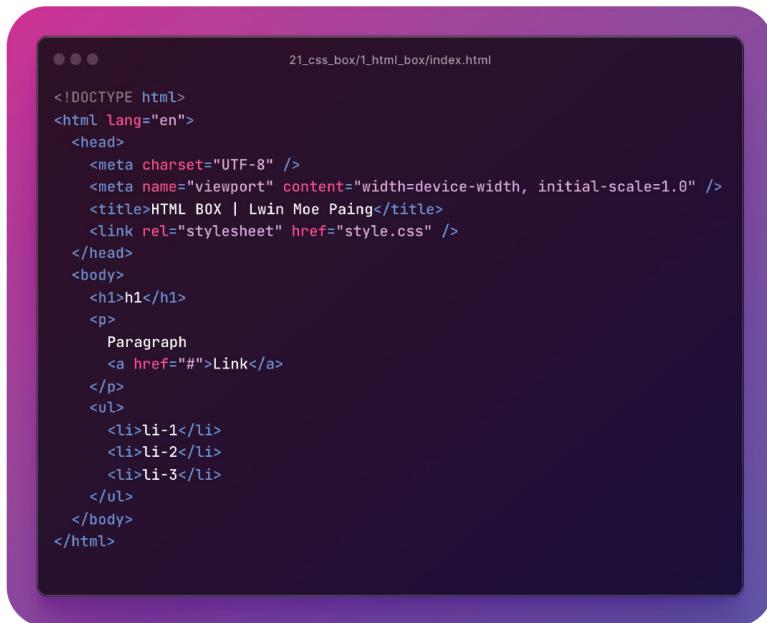
CSS နှင့် Boxes များ

HTML Element တိုင်းက သူတို့၏ ကိုယ်ပိုင် Box လေးတွေထဲမှာ နေထိုင်တယ်ဆိုတာ သိရအောင် ကျွန်ုတ်တို့ ပထမဆုံး အောက်ကအတိုင်း Code လေးတွေရေးပြီး စမ်းသပ်ကြည့်ရအောင်။

ပထမဦးစွာ ကျွန်ုတ်တို့ New Folder (ဖိုလ်ဒါအသစ်) တစ်ခု “21_css_box” ဆိုပြီးတည်ဆောက်လိုက်ပါမယ်။ ပြီးရင် အဲ ဖိုလ်ဒါအတွင်းမှာ ထပ်ပြီး နောက်ထပ် ဖိုလ်ဒါအသစ် “1_html_box” ဆိုပြီး တည်ဆောက်လိုက်ပါမယ်။ “1_html_box” ဆိုတဲ့ အတွင်းထဲမှာ “index.html” နဲ့ “style.css” ဆိုပြီး HTML နဲ့ CSS ဖိုင်(၂) ဂုဏ်ညွှန်ဆောက်လိုက်ပါမယ်။

index.html အတွက် Gist Code ကို ကူးယူလိုက်ပါ။

<https://gist.github.com/lwinmoepaing/1f65c77162a792b4e314a2fbee2f0f28>



```
21_css_box/1_html_box/index.html

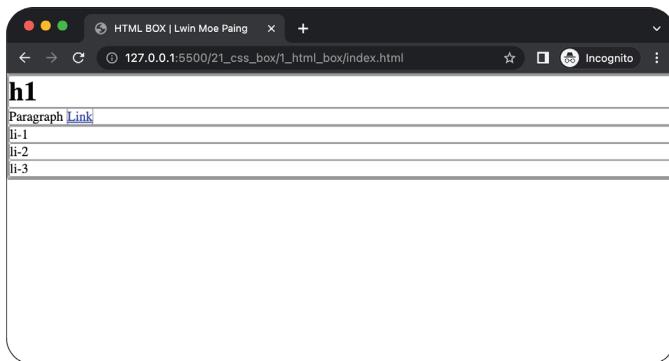
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>HTML BOX | Lwin Moe Paing</title>
    <link rel="stylesheet" href="style.css" />
</head>
<body>
    <h1>h1</h1>
    <p>
        Paragraph
        <a href="#">Link</a>
    </p>
    <ul>
        <li>li-1</li>
        <li>li-2</li>
        <li>li-3</li>
    </ul>
</body>
</html>
```

style.css အတွက် Gist Code ကို ကူးယူလိုက်ပါ။

<https://gist.github.com/lwinmoepaing/eb4da086ad3eb37173d675fde39de73d>



Browser ပေါ်စွင် အောက်က ပုံအတိုင်း မြင်ရပါမည်။

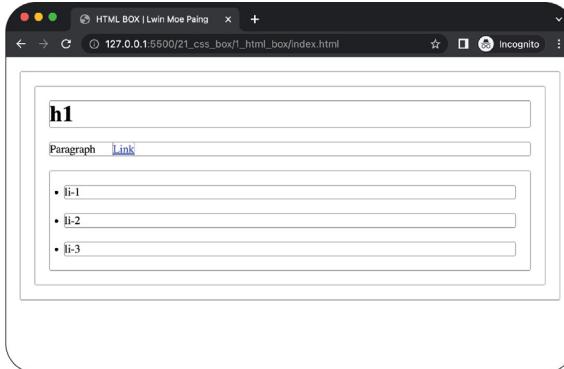


Browser မှာ <h1> <p> <a> နဲ့ elements များအားလုံးကို CSS Rules အောက် Border 1px နဲ့ မျဉ်းလိုင်းအရောင် #999999 Color သတ်မှတ်ပေးထားပါတယ်။ အခု Element တိုင်းမှာ Border လိုင်းတွေပေါ်နေပါပြီ။ style.css ထဲက Rule အောက် Margin နဲ့ Padding ကို 0px ပေးထားသောကြောင့် အားလုံးက Border မျဉ်းနဲ့ ကပ်နေတာကို သတိပြုလိုက်ပါ။

အခု style.css ထဲမှာ ဒီလိုပေးသားလိုက်ပြီး Browser ပေါ်စွင် Run ကြည်ပါ။

```
* {
    border: 1px solid #999999;
    margin: 20px;
    padding: 0px;
    border-radius: 3px;
```

margin တစ်ခုထက်ကိုသာ 20px ဖြောင်းလဲလိုက်သော အခါ အောက်က ပုံအတိုင်း Browser ပေါ်တွင်မြင်ရပါလိမ့်မယ်။

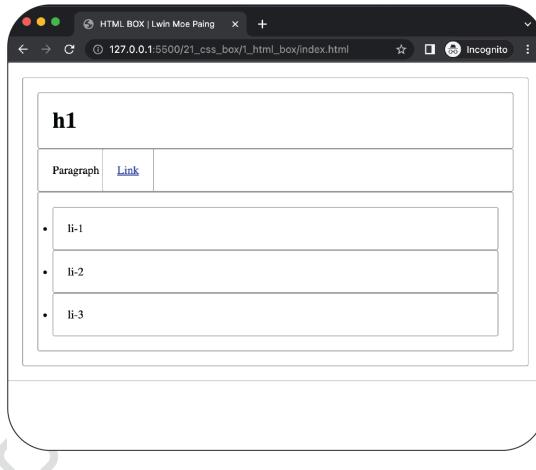


ပထမဆုံး လိုင်းသည် <html> Element ရဲ့လိုင်းပါ သူက Margin က Border ရဲ့ အပြင်ကို ကန်လိုက်ပါတယ်။ ဒုတိယက <body> Element ရဲ့လိုင်းပါ။ အတွင်းက စုံး <h1> Element ရဲ့ Border လိုင်းရဲ့ အပြင်ကို စားထားခြင်းဖစ်တယ် ပြီးတော့ အတွင်းစာသားကတော့ ကပ်နေတုန်းပါပဲ။ ဆိုလိုချင်တာက အကုန်လုံးက ဘူး၌ Box လေးတွေအတိုင်း HTML element တွေက နေထိုင်ကြပါတယ်။ အခုနောက်တစ်ကိုမိ margin ကို 0px ပြန်ထားပြီး padding ကို 20px ဖြောင်းပြီး စမ်းကြည့်ရအောင်။

style.css ထဲတွင် -

```
* {
    border: 1px solid #999999;
    margin: 0px;
    padding: 20px;
    border-radius: 3px;
}
```

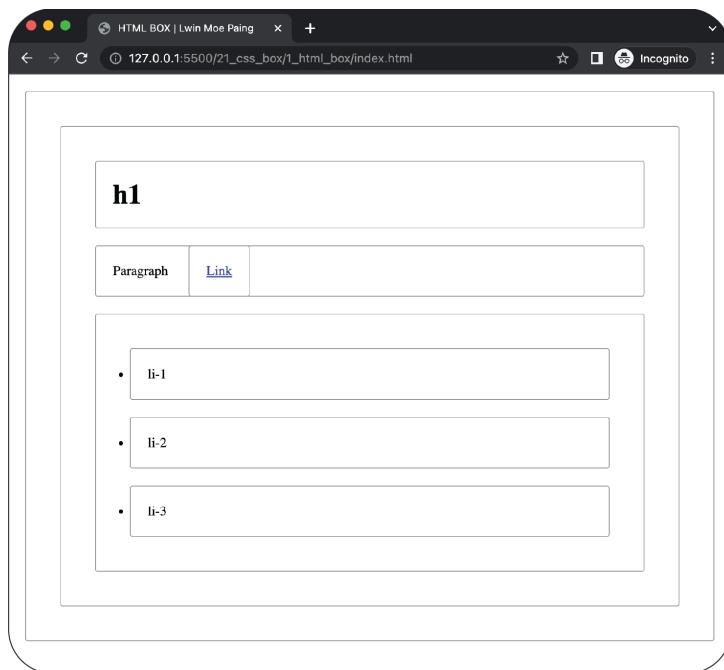
အောက်ပါပုံအတိုင်း Browser ပေါ်တွင်မြင်ရပါမယ်



padding တစ်ခုထဲကိုသာ 20px ပြောင်းလဲလိုက်သော Border အတွင်းပိုင်းကိုသာ Spacing ကို ခြားနားပေးသောကြောင့် အထက် စာသားတွေ နဲ့ မျဉ်းဟာ ကွာသွားမှာဖြစ်ပါ တယ်။ ဒါဆိုရင် အခုလို style.css ထဲမှာ ပြောင်းလဲရေးသားကြည့်လိုက်ပါ။

```
* {
    border: 1px solid #999999;
    margin: 20px;
    padding: 20px;
    border-radius: 3px;
}
```

Browser ပေါ်တွင် အောက်ပါ ပုံအတိုင်း Spacing ကို Border မျဉ်းလိုင်းရဲ့ အတွင်းကော အပြင်ဘက်ပါ ခြားသွားမည်ကို တွေ့ရပါမယ်။



ဒီအပိုင်းမှာ စာဖတ်သူတို့က Padding နဲ့ Margin အကြောင်းကို နားမလည်သေးလည်း ရပါသေးတယ် နောက်အခန်းတွေမှာ သေချာရှင်းပြသွားမှာပါ။ အခုက် Box လေးတွေနဲ့ HTML element တွေက သူ့အကန့်လေးနဲ့ သူရှိတာကို မြင်စေဖို့ပြုဖြစ်ပါတယ်။

အခုအကြောင်းအရာများကို CodePen မှာ သွားရောက် စမ်းသပ်နိုင်ပါတယ်။

<https://codepen.io/lwinmoeapaing/pen/NWEMoXG>

CSS Width များအကြောင်း

auto, **px**, **%** အစရိတ္တော် Unit တွန့် Box တွေရဲ့ Width Size (အကျယ်) တွေကို ဒီ
အပိုင်းမှာ သတ်မှတ်သွားမှာဖြစ်ပါတယ်။

"21_css_box" ဖိုလ်ဒါအတွင်းမှာ ထပ်ပြီး နောက်ထပ် ဖိုလ်ဒါအသစ် "2_width" ဆိုပြီး
တည်ဆောက်လိုက်ပါမယ်။ "2_width" ဆိုတဲ့ အတွင်းထဲမှာ "index.html" နဲ့ "style.css"
ဆိုပြီး HTML နဲ့ CSS ဖိုင် (J) ခုတည်ဆောက်လိုက်ပါမယ်။

အောက်က index.html အတွက် Gist Code ကို ကူးယူပေးပါ။

<https://gist.github.com/lwinmoeapaing/a62f1fa7935eb549dac50ff0f3c71963>

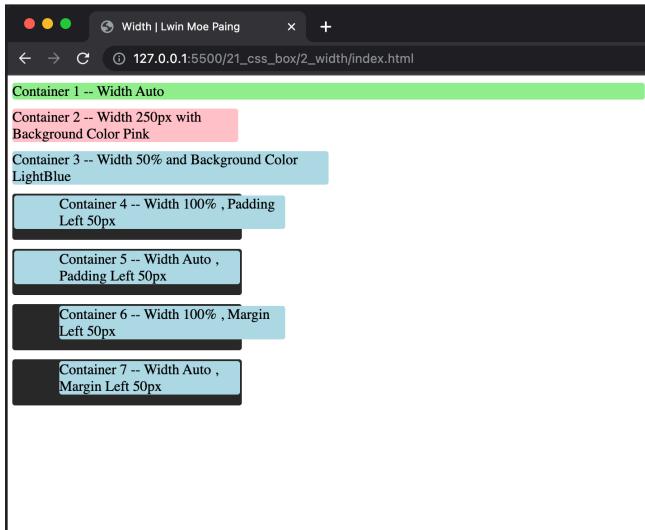
အောက်က style.css အတွက် Gist Code ကို ကူးယူပေးပါ။

<https://gist.github.com/lwinmoeapaing/bb1ea21ae3f4488279bca499793abf30>

အရင်ဆုံးအနေနဲ့ CodePen မှာ သွားရောက်စမ်းသပ်ကြည့်နိုင်ပါတယ်။

<https://codepen.io/lwinmoeapaing/pen/ZEmogLd>

အောက်ကပုံအတိုင်း ဘဏ်သူတို့ Browser မှာဖော်ပြပါလိမ့်မယ်။



နောက်တစ်မျက်နှာမှာ Code တွေကို ဖော်ပြသွားပါမယ်။

index.html အတွင်းက <body> Element အတွင်းရှိမယ့် Code ထွေကို အောက်ပါပဲ
အတိုင်းဖော်ပြထားပါတယ်။

```
● ● ● 21_css_box/2_width/index.html

<body>
    <!-- Start: Container-1 -->
    <div class="width-auto bg-lightgreen">Container 1 -- Width Auto</div>
    <!-- Finished: Container-1 -->
    <!-- Start: Container-2 -->
    <div class="width-250px bg-pink">
        Container 2 -- Width 250px with Background Color Pink
    </div>
    <!-- Finished: Container-2 -->
    <!-- Start: Container-3 -->
    <div class="width-50-percent bg-lightblue">
        Container 3 -- Width 50% and Background Color LightBlue
    </div>
    <!-- Finished: Container-3 -->
    <!-- Start: Container-4 -->
    <div class="width-250px container">
        <div class="width-100-percent bg-lightblue padding-left-50px">
            Container 4 -- Width 100% ,
            Padding Left 50px
        </div>
    </div>
    <!-- Finished: Container-4 -->
    <!-- Start: Container-5 -->
    <div class="width-250px container">
        <div class="width-auto bg-lightblue padding-left-50px">
            Container 5 -- Width Auto ,
            Padding Left 50px
        </div>
    </div>
    <!-- Finished: Container-5 -->
    <!-- Start: Container-6 -->
    <div class="width-250px container">
        <div class="width-100-percent bg-lightblue margin-left-50px">
            Container 6 -- Width 100% ,
            Margin Left 50px
        </div>
    </div>
    <!-- Finished: Container-6 -->
    <!-- Start: Container-7 -->
    <div class="width-250px container">
        <div class="width-auto bg-lightblue margin-left-50px">
            Container 7 -- Width Auto ,
            Margin Left 50px
        </div>
    </div>
    <!-- Finished: Container-7 -->
</body>
```

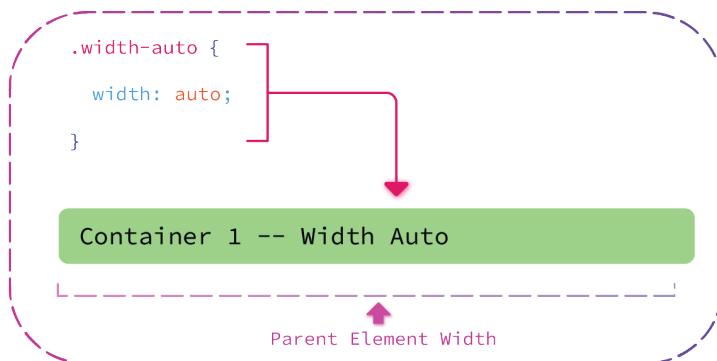
style.css အတွင်းရှိမယ့် Code တွေကို အောက်ပါပုံအတိုင်းဖော်ပြထားပါတယ်။

```
21_css_box/2_width/style.css

/* Container */
div {
    margin-bottom: 10px;
    border-radius: 3px;
}
.container {
    padding: 2px;
    background: #2b2b2b;
}
/* Background Colors */
.bg-pink {
    background-color: pink;
}
.bg-lightgreen {
    background-color: lightgreen;
}
.bg-lightblue {
    background-color: lightblue;
}
/* Widths */
.width-auto {
    width: auto;
}
.width-250px {
    width: 250px;
}
.width-50-percent {
    width: 50%;
}
.width-80-percent {
    width: 80%;
}
.width-100-percent {
    width: 100%;
}
/* Helpers */
.padding-left-50px {
    padding-left: 50px;
}
.margin-left-50px {
    margin-left: 50px;
}
```

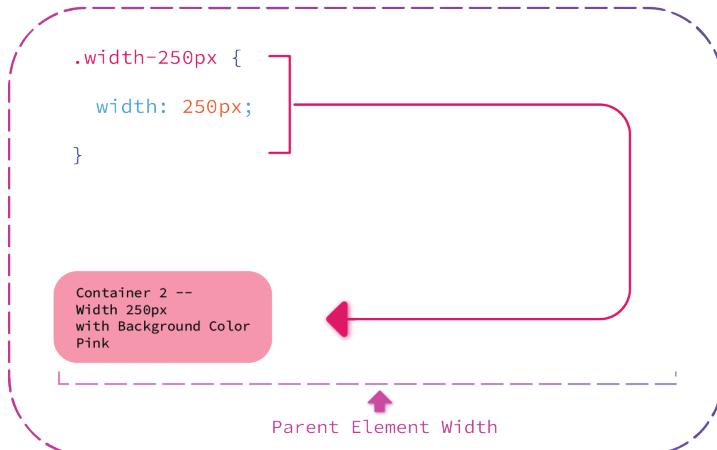
ဒါကိုတစ်ချက်အရင် လေ့လာကြည့်ရအောင် div တစ်ခုချင်းဆီကို margin bottom နဲ့ အောက်ဘက်ပိုင်းကို Spacing အကွာအဝေးမြှားထားခြင်လိုပါ။ .container Class Selector ကတော့ နောက်ခံအရောင် အနက်ရောင်နဲ့ အတွင်းသက် Spacing Padding 2px မြှားထားတာကို ပြင်ရမှာပါ။ .bg-pink, .bg-lightgreen, .bg-lightblue တို့ကတော့ Background Color တွေကို သတ်မှတ်ထားတာဖြစ်ပါတယ်။ /* Widths */ အောက်ပိုင်း CSS Rules တွေကတော့ ကျွန်ုတ်တို့ လေ့လာသွားမှာပါ။ /* Helper */ တွေကတော့ Padding (အတွင်း Spacing) နဲ့ Margin (အပြင် Spacing) တွေစိုးလို့ရအောင်ရေးထား ထားဖြစ်ပါတယ်။

Container - 1 ဆိတဲ့ အစိမ်းရောင်လေးနဲ့ Width Auto ကိုလေ့လာကြည့်ရအောင်။

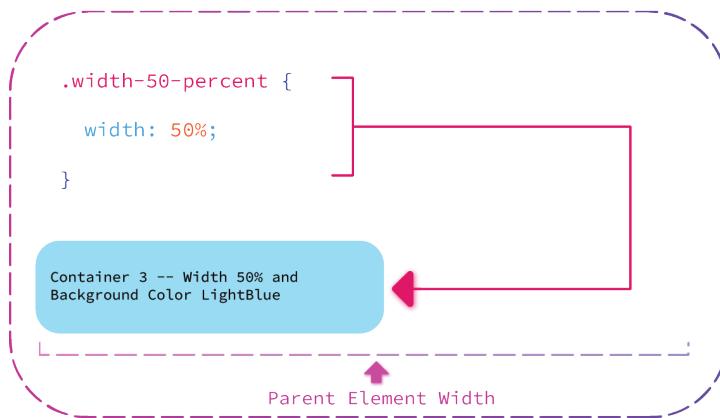


တန်ဖိုး အနေနဲ့ "auto" က Default မူလတန်ဖိုးပါ Block Element တွေအတွက် ကိုယ့်
ရဲ့ Parent Element ရဲ့ရှိသလောက် အကျယ်အတိုင်းယူသွားမှာပါ။ Block Element
ဆိတဲ့ကို စာဖတ်သူတို့က မမှတ်မိခဲ့ရင် HTML အခန်းက Generic Elements များ
ကို ပြန်လည်သွားဖတ်ပြီး မှတ်မိအောင်လုပ်နိုင်ပါတယ်။ Inline Element တွေအတွက်
ကတော့ ကိုယ်ပိုင် Content ရှိသလောက်သာယူသွားမှာပါ။ ဒီအပိုင်းမှာတော့ Block
Element တွေနဲ့ပဲ Example ပြလိုက်ပါမယ်။

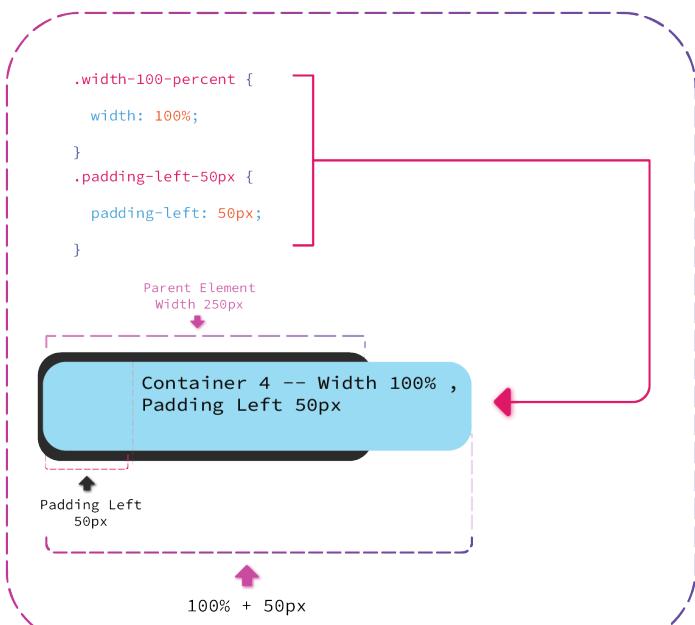
Container - 2 Width Px ဖြင့်ပေးခြင်းကိုလေ့လာကြည့်ရအောင်။ အောက်ကပုံအတိုင်း
250px ပေးလိုက်တဲ့အတွက် ထိုအကျယ်အတိုင်း ယူသွားတာဖြစ်ပါတယ်။



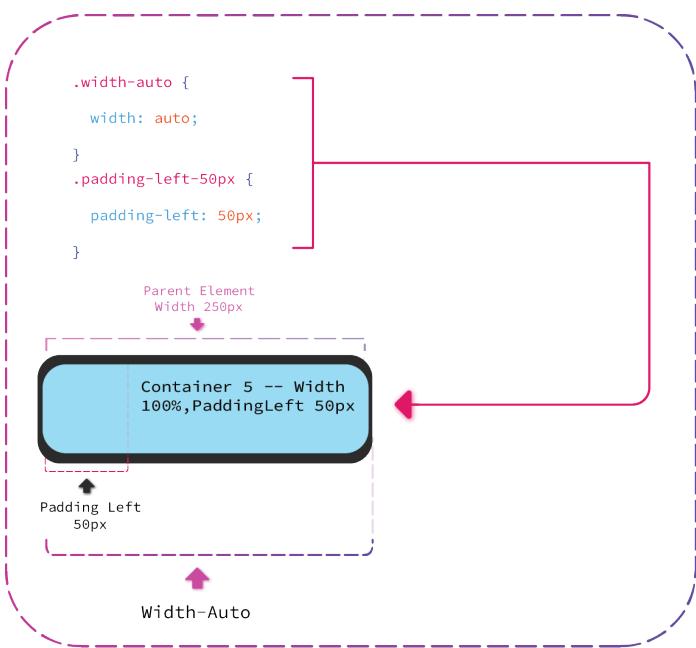
Container - 3 Width % (Percentage) ဖွင့်ပေးခြင်းကို အောက်ကပါတဲ့ လိုက်ပါ။



တန်ဖိုး 50% ထည့်လိုက်ပါတယ် အဲမှာပြောချင်တာက ဘယ်သူရဲ့ (50) ရာခိုင်နှုန်းလဲဆို ကိုယ့်ရဲ့ အပေါ်က Parent Element (မိဘ) ရဲ့ အကျယ်နဲ့ သက်ဆိုင်ပါတယ် Parent Element က Screen အပြည့်ဆိုရင် ကိုယ်ပိုင်အကျယ်က Screen ရဲ့ တစ်ဝက်ကို ရသွား မှာဖြစ်ပါတယ်။ ဥပမာ ကိုယ့်ရဲ့ အထက်က Parent Element အကျယ်က 300px ဆို ကိုယ့်ရဲ့ 50% သည် 150px ကိုရသွားမှာပါ။ Container 4 ကိုလေ့လာကြည့်ရအောင် ...။



Container 4 ရဲ့ Parent Element အကျယ်သည် 250px ဖြစ်ပြီး အနက်ရောက် Background နဲ့ ပါ။ Container-4 ကိုယ်တိုင်ကတော့ Padding Left (အတွင်း) Spacing 50 px ကိုခွာထားတာဖြစ်ပြီး ကိုယ်ပိုင် Width (100%) ကို ထည့်ထားတာဖြစ်ပါတယ်။ အဲတော့ တွက်ကြည့်ရအောင် width 100% ဆိုတော့ Parent ရဲ့အကျယ်အတိုင်း အပြည့် ယူထားတာပါ။ 250px အပြည့်ရသွားပါတယ် သို့ပေါ်ယော် Padding Left 50px နဲ့ ပေါင်းလိုက်တော့ 300px ဖြစ်သွားတာဖြစ်ပါတယ်။ အဲတော့ Parent အကျယ်ထက် ကျော် တွက်သွားပါတယ်။ ဒါဆိုရင် Container 5 ကို ဆက်လေ့လာကြည့်ရအောင် ...။



Container 5 လည်းပဲ Parent Element အကျယ်သည် 250px ဖြစ်ပြီး အနက်ရောက် Background နဲ့ ပါ။ Container-5 မှာလည်း Padding Left 50px ကိုထည့်သွင်းအသံ့ပြု ထားပါတယ်။ ထူးခြားချက်အနေနဲ့ သူကတော့ ဘောင်ကို ကျော်မထွက်သွားပါဘူး။ "auto" နဲ့ "percentage" တို့ရဲ့ ကွာခြားချက်ပါ။ Auto အနေနဲ့ ကိုယ်ပေးလိုက်တဲ့ အထဲ က Padding အကွာအဝေးကိုပါ သူရဲ့ အကျယ်ထဲမှာ Auto ချိန်ညံ့ပေးလိုက်ပါတယ်။

Container 6 နဲ့ Container 7 က လည်း အလားတူပါပဲ Auto နဲ့ Percentage (%) အသုံးပြုပုံကို စားဖတ်သူတို့ နားလည်အောင် ရေးသားဖော်ပြထားခြင်းဖြစ်ပါတယ်။ သူတို့ Container 6 နဲ့ 7 ကတော့ Margin အပြင် (Spacing) အကွာအဝေးကို အသုံးပြုထားတာပဲဖြစ်ပါတယ်။

အခြား **min-width** နဲ့ **max-width** ဆိုပြီးရှိပါသေးတယ်။ အောက်က ရေးသားနည်းကို တစ်ချက်ကြည့်ပေးပါ။

```
.min-width-200px {  
    min-width: 200px;  
}  
  
.max-width-200px{  
    max-width: 200px;  
}
```

min-width ရဲ့ အရှည်က Minimum Width ကို ကိုယ်စားပြုပါတယ် အနည်းဆုံး အကျယ်ထားမယ်လို့ ဆိုလိုခြင်းဖြစ်ပါတယ်။ ဥပမာ ထဲက **min-width: 200px** ဆိုတာက Selected လုပ်ထားတဲ့ Element ရဲ့ အနည်းဆုံး Width အကျယ် ကို 200px ထားမယ်လို့ ကြောာထားတာပါ။

max-width ရဲ့ အရှည်ကတော့ Maximum Width ကို ဆိုလိုတာဖြစ်ပါတယ် အများဆုံး အကျယ်ပေါ့ ဥပမာ ကျွန်ုတ်တို့ ထည့်လိုက်တဲ့ပုံက 1000px ကျော်နေတယ်ဆိုပါစို့ ပုံကအရမ်းကြီးနေသလိုဖြစ်နေလို့ အများဆုံးအကျယ် 200px ပဲလိုချင်တဲ့ အခါ ဒီလို သုံးလိုက်လိုပါတယ် **max-width: 200px** ပုံစံနဲ့ပါ။

auto, **px**, **%** အပြင် တခါတလေး **vw** ဆိုတဲ့ (Viewport Width) အရှည်ကောက် အနေနဲ့လည်း ရေးသားတာမျိုးတွေ့ကောင်းတွေ့ရပါတယ်။ 100vw ဆိုရင် 100 view port width (အကျယ်) ကိုဆိုလိုချင်တာဖြစ်ပြီး View Port ဆိုတာကတော့ Web Page တစ်ခုလုံးအတွက် visible area (မြင်နိုင်တဲ့) ဧရိယာကို ဖော်ပြတဲ့နေရာမှာသုံးပါတယ်။ အလွယ်ပြောရရင် Browser ရဲ့ အကျယ် 100vw ဆိုရင် Browser အကျယ် 100 ရာခိုင်နှစ်ဦးပါ။ 50vw ဆိုရင် Browser အကျယ်ရဲ့ တစ်ဝက် 50 ရာခိုင်နှံးလုံးသတ်မှတ်ပါတယ်။ ဒီလောက်ဆို ကျွန်ုတ်တို့ Width အကျယ်ကို သတ်မှတ်ပုံကိုလေ့လာမှတ်သားပြီးပါပြီ။

CSS Height များအကြောင်း

Width မှာ လေ့လာခဲ့သလိုပဲ **auto**, **px**, **%** အစရိတ္ထဲ Unit တွေနဲ့ Box တွေရဲ့ Height Size (အမြင်) တွေကို ဒီအပိုင်းမှာ သတ်မှတ်သွားမှာဖြစ်ပါတယ်။

“21_css_box” ဖိုလ်ဒါအတွင်းမှာ ထပ်ပြီး နောက်ထပ် ဖိုလ်ဒါအသစ် “3_height” ဆိုပြီး တည်ဆောက်လိုက်ပါမယ်။ “3_height” ဆိုတဲ့ Folder အတွင်းထဲမှာ “index.html” နဲ့ “style.css” ဆိုပြီး HTML နဲ့ CSS ဖိုင်(၂) ရုတည်ဆောက်လိုက်ပါမယ်။

အောက်က index.html အတွက် Gist Code ကို ကူးယူပေးပါ။

<https://gist.github.com/lwinmoepaing/29c872c156a9c877b6aa96f12eb72c48>

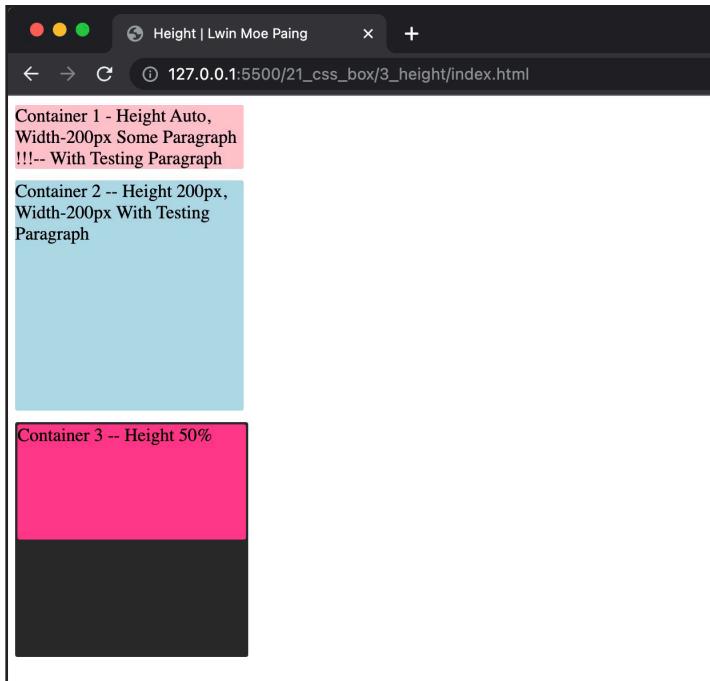
အောက်က style.css အတွက် Gist Code ကို ကူးယူပေးပါ။

<https://gist.github.com/lwinmoepaing/4f6c21c5c1b6f0c5b21377d6eef2359d>

အရင်ဆုံးအနေနဲ့ CodePen မှာ သွားရောက်စမ်းသပ်ကြည့်နိုင်ပါတယ်။

<https://codepen.io/lwinmoepaing/pen/eYQjrvx>

အောက်ကပုံအတိုင်း စာဖတ်သူတို့ Browser မှာဖော်ပြပါလိမ့်မယ်။



index.html အတွင်းရှိမယ့် Code တွေကို အောက်ပါပို့အတိုင်းဖော်ပြထားပါတယ်။

```
● ● ● 21_css_box/3.height/index.html

<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8" />
        <meta name="viewport" content="width=device-width, initial-scale=1.0" />
        <title>Height | Lwin Moe Paing</title>
        <link rel="stylesheet" href="style.css" />
    </head>
    <body>
        <!-- Start: Container-1 -->
        <div class="width-200px bg-pink height-auto">
            Container 1 - Height Auto, Width-200px
            Some Paragraph !!---
            With Testing Paragraph
        </div>
        <!-- Finished: Container-1 -->
        <!-- Start: Container-2 -->
        <div class="width-200px bg-lightblue height-200px">
            Container 2 -- Height 200px, Width-200px
            With Testing Paragraph
        </div>
        <!-- Finished: Container-2 -->
        <!-- Start: Container-3 -->
        <div class="width-200px height-200px container">
            <div class="bg-rose height-50-percent">
                Container 3 -- Height 50%
            </div>
        </div>
        <!-- Finished: Container-3 -->
    </body>
</html>
```

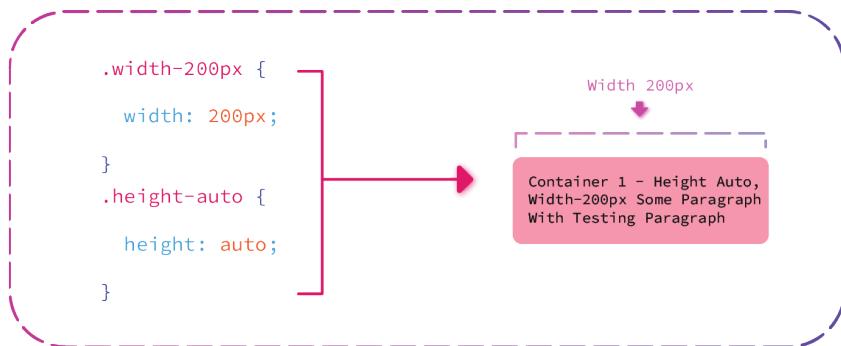
Width (အကျယ်) အခန်းမှာ လေ့လာခဲ့သလိုပဲ div တစ်ခုချင်းဆိုကို margin bottom နဲ့ အောက်ဘက်ပိုင်းကို Spacing အကွာအဝေးမြှားထားခြင်လိုပါ။ .container Class Selector ကတော့ နောက်ခံအရောင် အနက်ရောင်နဲ့ အတွင်းဘက် Spacing Padding 2px မြှားထားတာကို မြင်ရမှာပါ။ /* Heights */ အောက်ပိုင်း CSS Rules တွေကတော့ ကျွန်ုတ်တို့ လေ့လာသွားမှာဖြစ်ပါတယ်။

style.css အတွင်းရှိမယ့် Code တွေကို နောက်တစ်မျက်နှာတွင်ဖော်ပြထားပါမယ်။

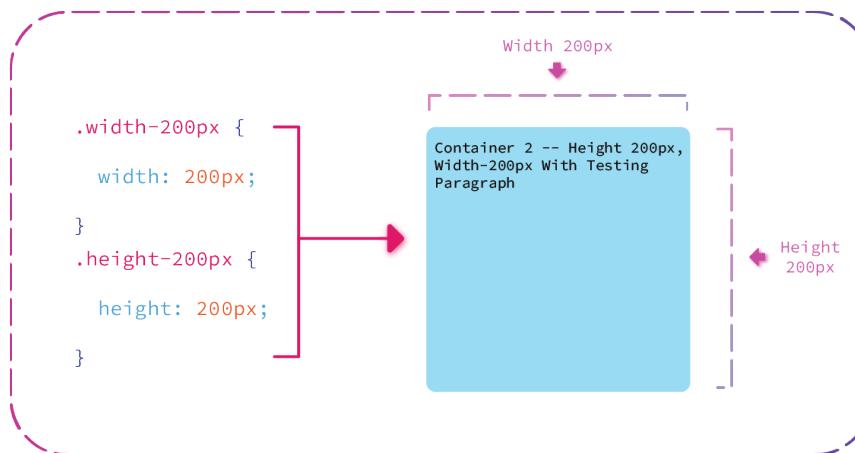
```
● ● ● 21_css_box/3_height/style.css

/* Container */
div {
    margin-bottom: 10px;
    border-radius: 2px;
}
.container {
    padding: 2px;
    background: #2b2b2b;
}
/* Background Colors */
.bg-pink {
    background-color: pink;
}
.bg-lightblue {
    background-color: lightblue;
}
.bg-rose {
    background-color: #ff3988;
}
/* Widths */
.width-200px {
    width: 200px;
}
/* Heights */
.height-auto {
    height: auto;
}
.height-200px {
    height: 200px;
}
.height-50-percent {
    height: 50%;
}
```

Container - 1 ဆိုတဲ့ ပန်းရောင်လေးနဲ့ Height Auto ကိုလေ့လာကြည့်ရအောင် ။

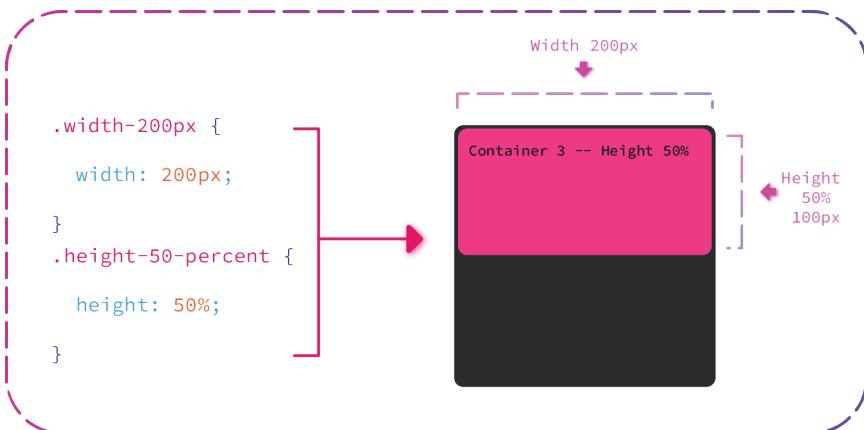


Height တန်ဖိုး အနေနဲ့ "auto" က Default မူလတန်ဖိုးပါ Element အတွင်းမှာရှိတဲ့ Content ရှိသလောက် Height (အမြင်) အတိုင်းယူသွားမှာပါ ။ ဥပမာ - Container 1 ထဲမှာရေးသားထားသော စာသားများ ရှည်ရွယ်ရှည်သလောက် အမြင်နေရာကို (auto) နေရာ ယူသွားမှာပါ ။



Height တန်ဖိုး Pixel တန်ဖိုးဖြင့်သတ်မှတ်ခြင်းက ရှင်းပါတယ် ။ ကိုယ်ပေးလိုက်တဲ့ Pixel အတိုင်းနေရာယူသွားမှာဖြစ်ပါတယ် ။

Container - 3 Width % (Percentage) ဖွင့်ပေးခြင်းကို အောက်ကပုံကိုကြည့်လိုက်ပါ။



Height တန်ဖိုး 50% ထည့်လိုက်ပါတယ် ဘယ်သူ့ရဲ့ (50) ရာခိုင်နှုန်းလဆို ကိုယ့်ရဲ့ အပေါ်က Parent Element (မိဘ) ရဲ့ Height(အမြင့်) နဲ့ သက်ဆိုင်ပါတယ်။ ဥပမာ ကိုယ့်ရဲ့ အထက်က Parent Element Height က 200px ဖြစ်ဖော်ပါတယ်။ ဆိုတော့ အခု Element Height 50% သည် တစ်ဝက်ဖြစ်လို့ 100px ကိုရသွားမှာပါ။

အခြား **min-height** နဲ့ **max-height** ဆိုပြီးရှိပါသေးတယ်။ အောက်က ရေးသားနည်းကို တစ်ချက်ကြည့်ပေးပါ။

```

    .min-height-200px {
        min-height: 200px;
    }
    .max-height-200px{
        max-height: 200px;
    }
  
```

min-height ရဲ့ အရှည်က Minimum Height ကို ကိုယ်စားပြုပါတယ်။ အနည်းဆုံး အမြင့် ထားမယ်လို့ ဆိုလိုခြင်းဖြစ်ပါတယ်။ ဥပမာ **min-height: 200px** ။

max-height ရဲ့ အရှည်က Maximum Height ကို ကိုယ်စားပြုပါတယ်။ အများဆုံး အမြင့် ထားမယ်လို့ ဆိုလိုခြင်းဖြစ်ပါတယ်။ ဥပမာ **max-height: 200px** ။

auto, **px**, **%** အပြင် တရာ့တလေ **vh** ဆိုတဲ့ (Viewport Height) အရှည်ကောက် အနေနဲ့လည်း ရေးသားတာမျိုးတွေ၊ ကောင်းတွေ၊ ရပါလိမ့်မယ်။ ဥပမာ - 100vh ဆိုရင် Screen ရုံးသလောက်အမြင့်ကို ယူလိုက်မှာပါ ဖြစ်ပါတယ်။

CSS Overflowing Contents များအကြောင်း

"21_css_box" ဖိုလ်ဒါအတွင်းမှာ ထပ်ပြီး နောက်ထပ် ဖိုလ်ဒါအသစ် "4_overflow" ဆိုပြီး တည်ဆောက်လိုက်ပါမယ်။ "4_overflow" ဆိုတဲ့ အတွင်းထဲမှာ "index.html" နဲ့ "style.css" ဆိုပြီး HTML နဲ့ CSS ဖိုင်(၂) ခုတည်ဆောက်လိုက်ပါမယ်။

အောက်က index.html အတွက် Gist Code ကို ကူးယူပေးပါ။

<https://gist.github.com/lwinmoepaing/b64566769a061a5dc4ae425e93a24e5b>

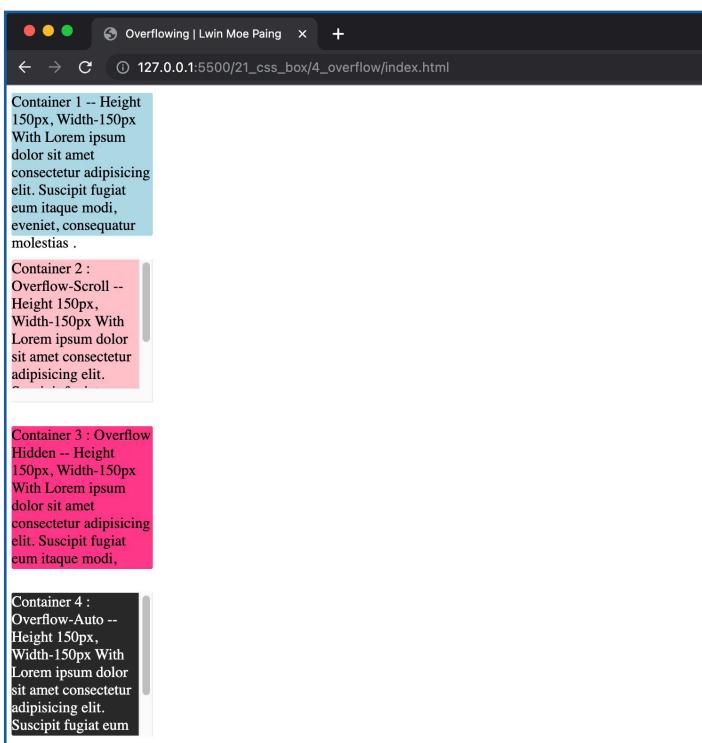
အောက်က style.css အတွက် Gist Code ကို ကူးယူပေးပါ။

<https://gist.github.com/lwinmoepaing/0dfcd2ae00003781c99213eda9f78766>

အရင်ဆုံးအနေနဲ့ CodePen မှာ သွားရောက်စင်းသပ်ကြည့်စိုင်ပါတယ်။

<https://codepen.io/lwinmoepaing/pen/dyQjjza>

အောက်ကပုံအတိုင်း တဖတ်သူတို့ Browser မှာဖော်ပြပါလိမ့်မယ်။



index.html အတွင်းရှိမယ့် Code တွေကို အောက်ပါပုံအတိုင်းဖော်ပြထားပါတယ်။

```
● ● ● 21_css_box/4_overflow/index.html

<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8" />
        <meta name="viewport" content="width=device-width, initial-scale=1.0" />
        <title>Overflowing | Lwin Moe Paing</title>
        <link rel="stylesheet" href="style.css" />
    </head>
    <body>
        <!-- Start: Container-1 -->
        <div class="width-150px height-150px bg-lightblue">
            Container 1 -- Height 150px, Width-150px With Lorem ipsum dolor sit amet
            consectetur adipisicing elit. Suscipit fugiat eum itaque modi, eveniet,
            consequatur molestias .
        </div>
        <!-- Finished: Container-1 -->
        <!-- Start: Container-2 -->
        <div class="overflow-scroll width-150px height-150px bg-pink">
            Container 2 : Overflow-Sroll -- Height 150px, Width-150px With Lorem
            ipsum dolor sit amet consectetur adipisicing elit. Suscipit fugiat eum
            itaque modi, eveniet, consequatur molestias .
        </div>
        <!-- Finished: Container-2 -->
        <!-- Start: Container-3 -->
        <div class="overflow-hidden width-150px height-150px bg-rose">
            Container 3 : Overflow Hidden -- Height 150px, Width-150px With Lorem
            ipsum dolor sit amet consectetur adipisicing elit. Suscipit fugiat eum
            itaque modi, eveniet, consequatur molestias .
        </div>
        <!-- Finished: Container-3 -->
        <!-- Start: Container-4 -->
        <div class="overflow-auto width-150px height-150px bg-dark text-white">
            Container 4 : Overflow-Auto -- Height 150px, Width-150px With Lorem
            ipsum dolor sit amet consectetur adipisicing elit. Suscipit fugiat eum
            itaque modi, eveniet, consequatur molestias .
        </div>
        <!-- Finished: Container-4 -->
    </body>
</html>
```

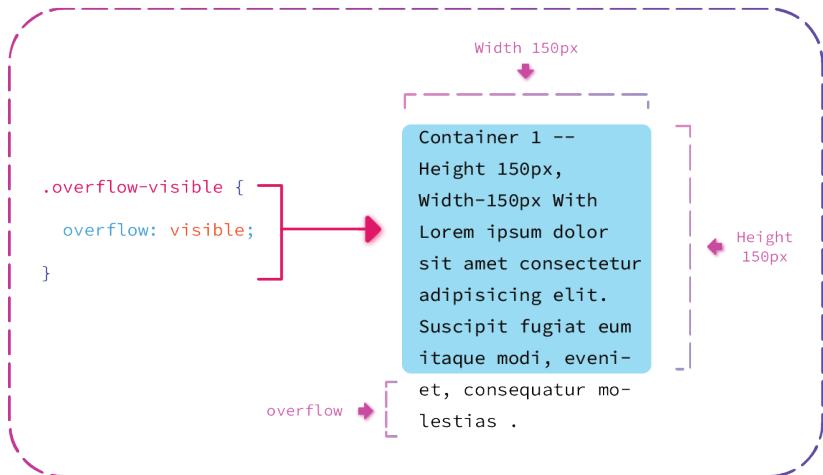
/* Overflow */ အောက်ပါင်း CSS Rules တွေကတော့ ကျွန်ုတ်တို့ လေ့လာသွားမှာ
ဖြစ်ပါတယ်။

style.css အတွင်းရှိမယ့် Code တွေကို အောက်ပါပုံအတိုင်းဖော်ပြထားပါတယ်။

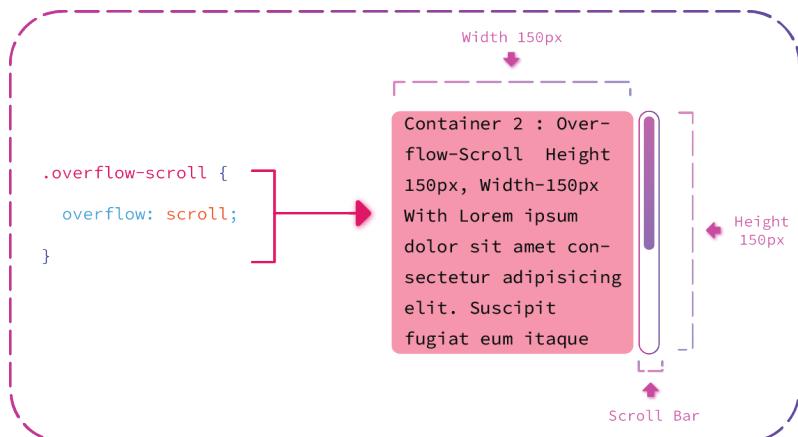
```
21_css_box/4_overflow/index.html

/* Container */
div {
    margin-bottom: 25px;
    border-radius: 2px;
}
.container {
    padding: 2px;
    background: #2b2b2b;
}
/* Background Colors */
.bg-pink {
    background-color: pink;
}
.bg-lightblue {
    background-color: lightblue;
}
.bg-rose {
    background-color: #ff3988;
}
.bg-dark {
    background-color: #2b2b2b;
}
/* Text */
.text-white {
    color: white;
}
/* Overflows */
.overflow-scroll {
    overflow: scroll;
}
.overflow-hidden {
    overflow: hidden;
}
.overflow-auto {
    overflow: auto;
}
/* Widths */
.width-150px {
    width: 150px;
}
/* Heights */
.height-150px {
    height: 150px;
}
```

Container - 1 ကိုကြည့်လိုက်ပါ အောက်က ပုံလေးထဲမှာရှိပါတယ်။ သူ့အထဲမှာ overflow နှုပါတ်သတ်တဲ့ class တစ်ခုမှုမသုံးထားပါဘူး။ Element တိုင်းမှာ ပင်ကို တန်ဖိုး overflow visible အဖြစ်ပါပါတယ်။ အရင်ဆုံး overflow ဆိုတာက သတ်မှတ်ထားတဲ့ Area (ဧရိယာ) ထက်ကျော်လွန်သွားတဲ့ အခြေအနေကို ဆိုလိုတာပါ။

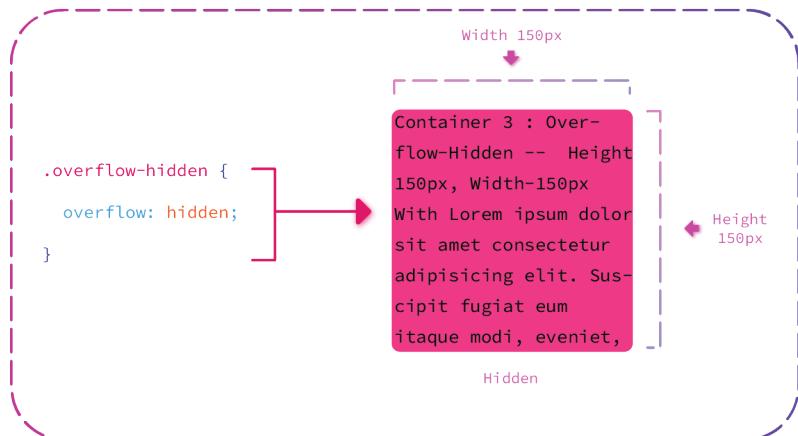


အခု Container မှာဆိုရင် Width 150px နဲ့ Height 150px ဆိုပြီးပေးထားတာပါ။ အထဲ က Content စာသားကတော့ သတ်မှတ်ထားတဲ့ Width နဲ့ Height ထက်ပိုသွားတဲ့အခါ Overflow ဖြစ်သွားပါတယ်။ အပြင်ကို ထွက်သွားတာကို မြင်ရမှာပါ။ Element တိုင်းမှာ overflow: visible ဆိုပြီး ပင်ကိုတန်ဖိုးရှိတဲ့အတွက် ကြောင့် ထွက်သွားတာဖြစ်ပါတယ်။ ဒီလို အခြေအနေမျိုးကို ထိန်းဖို့အတွက် အခြား overflow တန်ဖိုးတွေကို လေ့လာကြည့်ရအောင် ။ အောက်က overflow: scroll အသုံးပြုပုံကို လေ့လာကြည့်ပါမယ်။

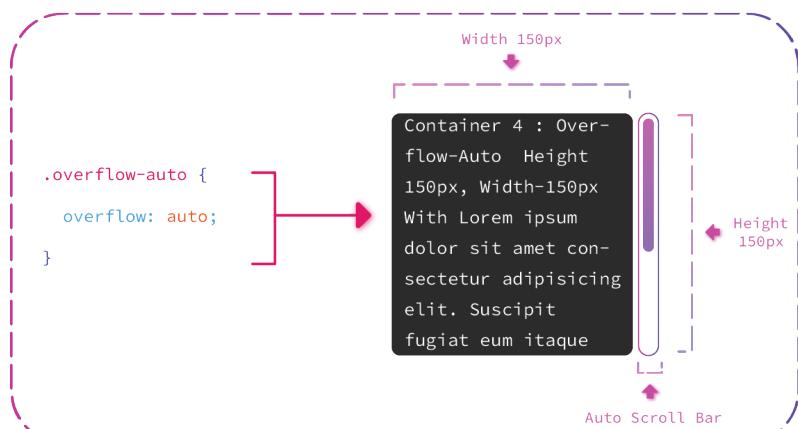


Container 2 မှာဆိုရင် overflow ရဲ့ တန်ဖိုးကို "scroll" ဆိုပြီး အသုံးပြုထားတာကိုတွေ့ရှုရအောင် ဖြစ်ပါတယ်။ အဓိပ္ပာတ်က Scroll ဆွဲလို့ရအောင် ဖြစ်ပါတယ်။ ညာဘက်မှာလဲ Scroll Bar လေး ထွက်ပေါ်လာတာကိုမြင်ရမှာပါ။ အောက်ကို ခွဲချေလိုက်ရင် လိုအပ်တဲ့ တာသားတွေကို Scroll Bar ကိုအသုံးပြုပြီးမြင်ရမှာဖြစ်ပါတယ်။

ဒါဆိုနောက်တစ်မျိုး hidden ကိုလေ့လာကြည့်ရအောင်။ သူကျတော့ Scroll လိုလည်း နောက်ထပ် Scroll Bar ကြီးနဲ့မဖော်ပြစ်ချင်ဘူး၊ နောက် ကျော်လွန်သွားတဲ့ Content တွေကိုလည်း မဖော်ပြချင်တဲ့အပါ Hidden လုပ်လိုက်လို့ရပါတယ်။ ဒါကို အောက်က ပုံ ကတစ်ဆင့်လေ့လာကြည့်ရအောင် ...။



အဲတော့ ကျော်လွန်သွားတဲ့ တာသားတွေကို Hidden (ဖုံးကွယ်) လိုက်တာကို မြင်တွေ့ရမှာဖြစ်ပါတယ်။ နောက်တစ်မျိုးကျွန်ုပ်ပေးတယ် "auto" ဖြစ်ပါတယ် သူကျတော့ မလိုအပ်ရင်တော့ Scroll မပေါ်ဘူး။ Content ကျော်လွန်သွားတဲ့ အခြေအနေမှာဆိုရင် တော့ Scroll Bar နဲ့ပြုလုပ်ပေးလိုက်မှာဖြစ်ပါတယ်။ အောက်ကပုံကို လေ့လာကြည့်ပါ။



အခုပုံမှာ Container 4 အတွက် စာဖတ်သူတို့က စာသားတွေကို လျော့လိုက်ပြီးတော့ ရေးသားလိုက်ပါက Scroll Bar မပေါ်ပါ ပျောက်ကွယ်သွားမှာဖြစ်ပါတယ်။ ကိုယ့်ရဲ့ Browser မှာစမ်းသပ်ရေးသားကြည့်နိုင်ပါတယ်။ ထိုအပြင် သီးခြား Horizontal အတွက် overflow ကိုပဲ ထိန်းချုပ်ခြင်ရင် (သို့မဟုတ်) Vertical အမြင့်အတွက်ပဲ သီးသန်း overflow ကို သီးခြား ရေးသားနိုင်ပါသေးတယ်။ ဒေါက်ကရေးသားနည်းကို လေ့လာပေးပါ။

```
.overflow-x-scroll {           .overflow-y-auto {  
    overflow-x: scroll;        overflow-y: auto;  
}  
}
```

အခုဆိုရင် overflow-x ဆိုတာကတော့ အလျားလိုက် ကျော်လွန်သွားတာကိုပဲ ထိန်းချုပ်တာဖြစ်ပါတယ်။ overflow-y ကတော့ ထောင်လိုက် အမြင့်တွေမှာ Content တွေ ကျော်လွန်သွားတာကို ထိန်းချုပ်ချင်တဲ့အခါ လိုသလိုအသုံးပြုနိုင်ပါတယ်။

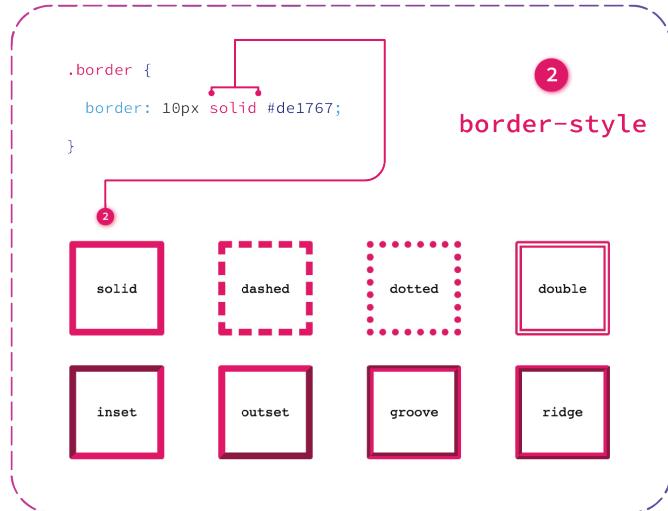
CSS Borders များအကြောင်း

ကျွန်တော်တို့ ဘောင်လိုင်းမျဉ်းကြောင်းများကို border property ကိုအသုံးပြုပြီး ရေးသားလိုပါတယ်။

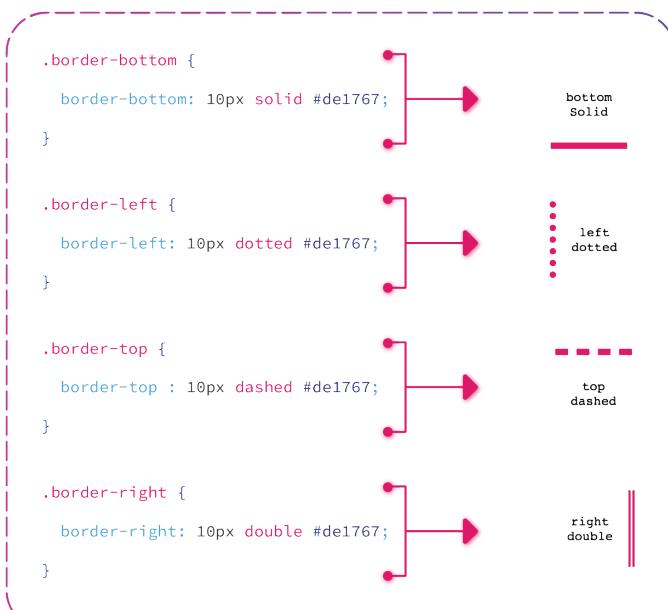
```
.border {  
    border: 10px solid #de1767;  
}  
  
border-width  
border-style  
border-color
```

border: ရဲ့အနောက်မှာ ပထမ တစ်ခုက border-width ဖြစ်ပါတယ် မျဉ်းလိုင်းရဲ့အထူကို သတ်မှတ်ရတာပါ။ ဒုတိယ တစ်ခုက border-style ပါ မျဉ်းခဲ့၊ ပုံစံကိုသတ်မှတ်ပေးရတာ ဖြစ်ပြီး နောက်ဆုံး တစ်ခုက border-color အရောင်ကိုသတ်မှတ်ပေးရပါတယ်။ ဒီရေးသားနည်းကတော့ အသုံးများတဲ့ အတိုကောက်ရေးသားနည်းဖြစ်ပါတယ်။

border-style နေရာမှာ အောက်က ပုံအတိုင်း စာဖတ်သူတို့က တန်ဖိုးတွေကို ထည့်သွင်း
အသုံးပြုနိုင်ပါတယ်။ သို့ပေမယ့် ပထမ အတန်းက solid, dashed, dotted, double
တန်ဖိုးများသာ အသုံးများပါတယ်။

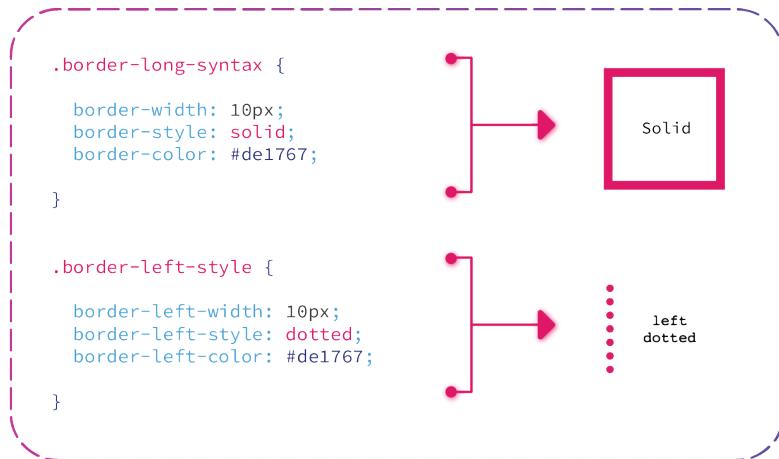


border-width တန်ဖိုးကိုတော့ 10px ကနေ အခြား တန်ဖိုးတွေ ပြောင်းလဲပြီး စာ
ဖတ်သူတို့က စမ်းပြီးရေးသားနိုင်ပါတယ်။ အနောက်က border-color ကိုလည်း
အလားတူစွာ ပြောင်းလဲပြီး အရောင်များစမ်းနိုင်ပါတယ်။ အခုနောက်ထပ် မျဉ်းလိုင်း
တစ်ဖက်တစ်ချက်စိကိုပဲ Style ပေးတဲ့ ပုံစံကိုလေ့လာကြည့်ရအောင်။



Value ရေးသားနည်းကတော့ နိုင်ပုံစံအတိုင်းပဲ border-width border-style border-color ဖြစ်ပါတယ်။ Property Name ရဲ့အနေက်မှာတော့ position (င့်) နေရာကို ထည့်ပြီး ရေးသားနိုင်ပါတယ်။ border-top, border-right, border-bottom, border-left ဆိုပြီး ကိုယ်သတ်မှတ်ပေးချင်တယ့် အရပ်ကိုထည့်ပြီး ရေးသားနိုင်ပါတယ်။

အရှည်ကောက်ရေးသားနည်းကို အောက်ကပုံမှာ လေ့လာကြည့်ရအောင်။ ဒီလိုပုံစံက စာဖတ်သုတိအတွက် အရှည်ကောက် ရေးသားနည်းရှိတယ်ဆိုသာ သိအောင် သာဖော်ပြလိုက်ရတာဖြစ်ပါတယ်။ လက်တွေ့မှာ အသုံးနည်းပါတယ်။



"21_css_box" ဖိုလ်ဒါအတွင်းမှာ ထပ်ပြီး နောက်ထပ် ဖိုလ်ဒါအသစ် "5_border" ဆိုပြီး တည်ဆောက်လိုက်ပါမယ်။ "5_border" ဆုတော့ အတွင်းထဲမှာ "index.html" နဲ့ "style.css" ဆိုပြီး HTML နဲ့ CSS ဖိုင်(J) ခုတည်ဆောက်လိုက်ပါမယ်။

အောက်က index.html အတွက် Gist Code ကို ကူးယူပေးပါ။

<https://gist.github.com/lwinmoepaing/c6bf100ae1a3c6ef0f07ee76dceb1d18>

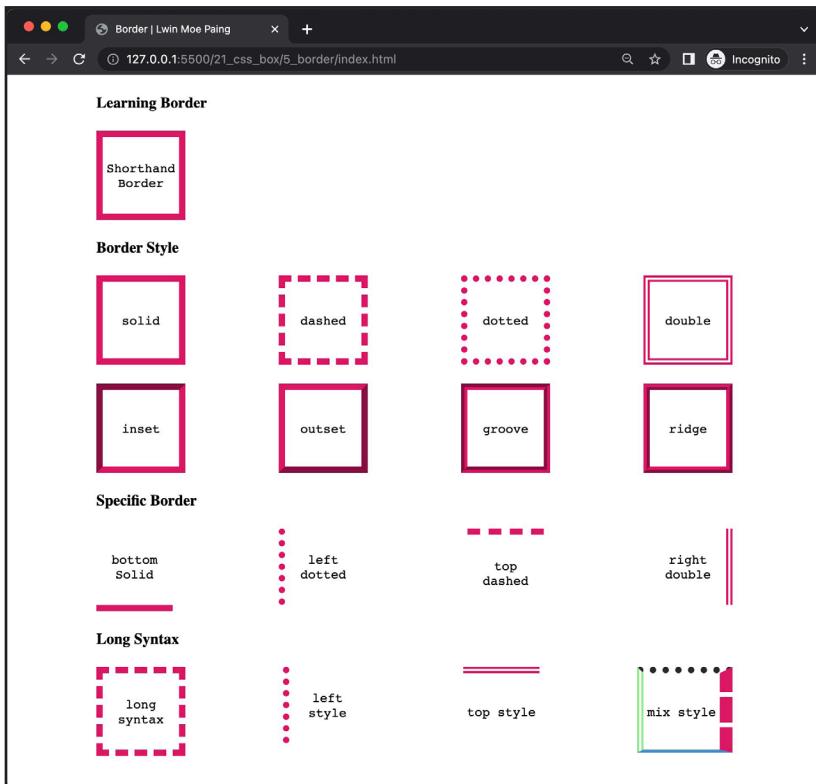
အောက်က style.css အတွက် Gist Code ကို ကူးယူပေးပါ။

<https://gist.github.com/lwinmoepaing/d488cf69a84c285b6a735845f860ba49>

အရင်ဆုံးအနေနဲ့ CodePen မှာ သွားရောက်စမ်းသပ်ကြည့်နိုင်ပါတယ်။

<https://codepen.io/lwinmoepaing/pen/qBQMPQX>

ပေးထားသော Code များကို စာဖတ်သူတို့ Browser မှာ Run လိုက်တဲ့ အခါ အောက်ပ ပုံအတိုင်း ရလာမှာဖြစ်ပါတယ်။ style.css ထဲတွင်ရေးသားထားသော Rule များကို လေ့လာကြည့်ပါက စာဖတ်သူတို့ လေ့လာထားခဲ့ပြီးဖြစ်တဲ့ border style များအကြောင်းတွေပ ဖြစ်ပါတယ်။



နောက် တစ်မျက်နှာမှာ ကျွန်ုတ်တို့ Border-Radius အကြောင်းဆက်လက်လေ့လာသွားပါမယ်။

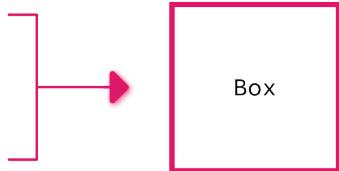
CSS Border Radius အကြောင်း

border ဘောင်မျဉ်းလိုင်းတွေဖော်ပြခြင်းကို လေ့လာပြီးပါပြီ။ အခု ဘောင်ရဲ့ ထောင့်စွန်းလေးတွေကို Radius သုံးပြီး ကျွေးသွားအောင် ပြုလုပ်ကြပါမယ်။

အရင်ဆုံး ကျွန်ုပ်တော်တို့ Border 3px နဲ့အတူ အကျယ် Width 100px နဲ့ အမြင့် Height 100px ရှိတဲ့ Box တစ်ခုပြုလုပ်လိုက်မယ်။ ပြီးရင် အောက်က ပုံထဲကအတိုင်း အသုံးပြု ပုံကို လေ့လာကြည့်ရအောင်။

အရင်ဆုံး Width & Height 100px ရှိတဲ့ Box တစ်ခု တည်ဆောက်လိုက်ပါမယ်။

```
.radius-box {
    width: 100px;
    height: 100px;
    border: 3px solid #de1767;
}
```

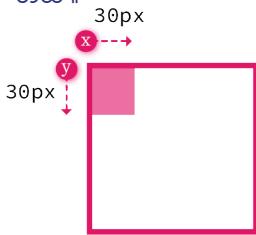


border-radius: 30px ပေးလိုက်တဲ့အခါ ညာဘက်က Box အတိုင်း ထောင့်စွန်းများကော် သွားမှာဖြစ်ပါတယ်။

```
.radius-box {
    width: 100px;
    height: 100px;
    border: 3px solid #de1767;
    border-radius: 30px;
}
```



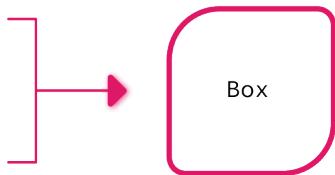
ထောင့်ကနေစွဲပြီး X ဝန်ရှိုးအတွက် 30px နဲ့ Y ဝန်ရှိုးအတွက် 30px အထိယူလိုက်ပါ တယ်။ အဲကြေားထက် လေးထောင့်ကွက်လေးအတွင်းကို ကျွေးပိုက်ပေးလိုက်ခြင်းဖြစ်ပါ တယ်။



border-radius: 30px/15px; ဆိတ် ပုံစံနဲ့ လည်းရေးသားလို့ရပါတယ်။ x အတွက် 30px ယူပြီး y အတွက် 15px နေရာယူသွားပြီး ကျွေးပေးလိုက်မှာဖြစ်ပါတယ်။

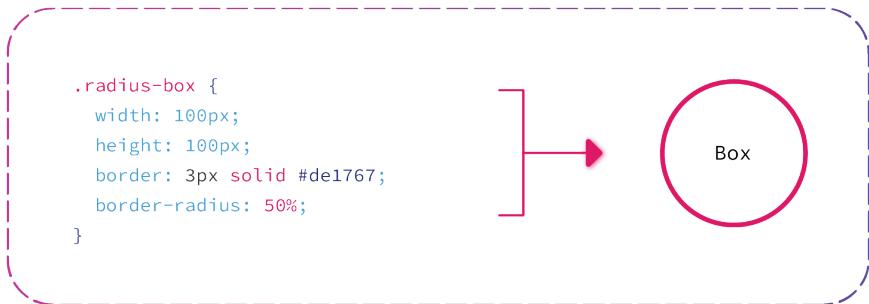
Border Radius Value ကို နှစ်မျိုးတည်းရေးသားနည်းနဲ့လည်း ကွေးတာမျိုးရှိပါသေး တယ် ဒါကို အောက်ကပုံမှာ လေ့လာကြည့်ရအောင် ...!!

```
.radius-box {
    width: 100px;
    height: 100px;
    border: 3px solid #de1767;
    border-radius: 30px 15px;
}
```



}

Border Radius Value ကို Percentage ရာစိုင်နှုန်းပေးပြီးလည်းရေးသားနိုင်ပါတယ်။ သို့ပေါ့မယ် သတိထားရမှာက အောက်ကပုံထက် 50% သည် width အကျယ်ရဲ့ 50% နဲ့ height အမြင့်ရဲ့ 50% နှုန်းကိုယူပြီး ကွေးလိုက်သွားမှာပါ။ အဲအတွက် Width နဲ့ Height ညီနေတဲ့ စတုရန်းမှသာ စက်ပိုင်းပုံစံကိုရမှာပါ အသုံးပြုမှုပုံစံများပါတယ်။



Border Radius အကြောင်းကို စာဖတ်သူတို့ လေ့လာခဲ့ပြီးဖြစ်ပါတယ်။

အဲတော့ Coding ပိုင်းအရ “21_css_box” ဖိုလ်ဒါအတွင်းမှာ နောက်ထပ် ဖိုလ်ဒါ အသစ် “6_border_radius” ဆိုပြီး တည်ဆောက်လိုက်ပါမယ်။ “6_border_radius” ဆိုတဲ့ အတွင်းထဲမှာ “index.html” နဲ့ “style.css” ဆိုပြီး HTML နဲ့ CSS ဖိုင်(၂) ခု တည်ဆောက်လိုက်ပါမယ်။

စာဖတ်သူတို့က index.html တွင် ဒီအောက်က Code ကိုကူးယူပြီးစမ်းကြည့်လို့ရပါမယ်။

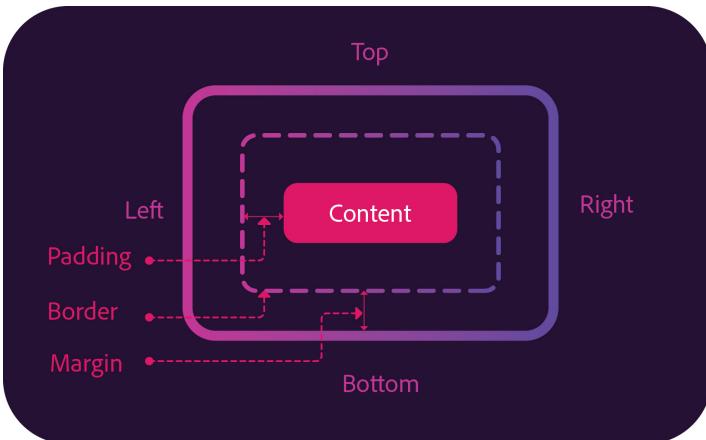
<https://gist.github.com/lwinmoepaing/9160d56c513f7794c0cf0708835062e7>

style.css တွင် ဒီအောက်က Code ကိုကူးယူပြီးစမ်းကြည့်ပါ။

<https://gist.github.com/lwinmoepaing/bb9079562deb04d426c7f2c3ce207a3f>

Codepen မှာလည်းတိုက်ရှိက် စမ်းကြည့်နိုင်ပါတယ်။

<https://codepen.io/lwinmoepaing/pen/zYMMRVG>



နောက် အခန်းတွေမှာ Padding နဲ့ Margin ကို လေ့လာသွားကြဖို့ဖြစ်ပါတယ်။ HTML Element တိုင်းသည် သူရဲ့ ကိုယ်ပိုင် Box လေးတွေထဲမှာနေထိုင်ပါတယ်။ Border ပေးလိုက်ရင် Border Line လေးမြင်ရ ပြီး Box သမ္မတလေးဖြစ်နေမှာပါ။

Border အတွင်းဘက်ကို အကွာအဝေး ခြားချင်တဲ့အခါ Padding သုံးပြီး၊ အပြင်ဘက် ကို ခြားချင်တဲ့အခါ Margin သုံးပါတယ်။

Position အရ Top, Left, Right, Bottom နဲ့လည်းခွဲ့လို့ရပါသေးတယ်။

CSS Margin အကြောင်း

“21_css_box” ဖိုလ်ဒါအတွင်းမှာ ဖိုလ်ဒါအသစ် “7_margin” ဆိုပြီး တည်ဆောက်လိုက်ပါမယ်။ “7_margin” အတွင်းထဲမှာ “index.html” နဲ့ “style.css” ဆိုပြီး HTML နဲ့ CSS နိုင်(၂) ခုတည်ဆောက်လိုက်ပါမယ်။

အောက်က index.html အတွက် Gist Code ကို ကူးယူပေးပါ။

<https://gist.github.com/lwinmoepaing/ba02ee24999abc6f3baec939c5e967b4>

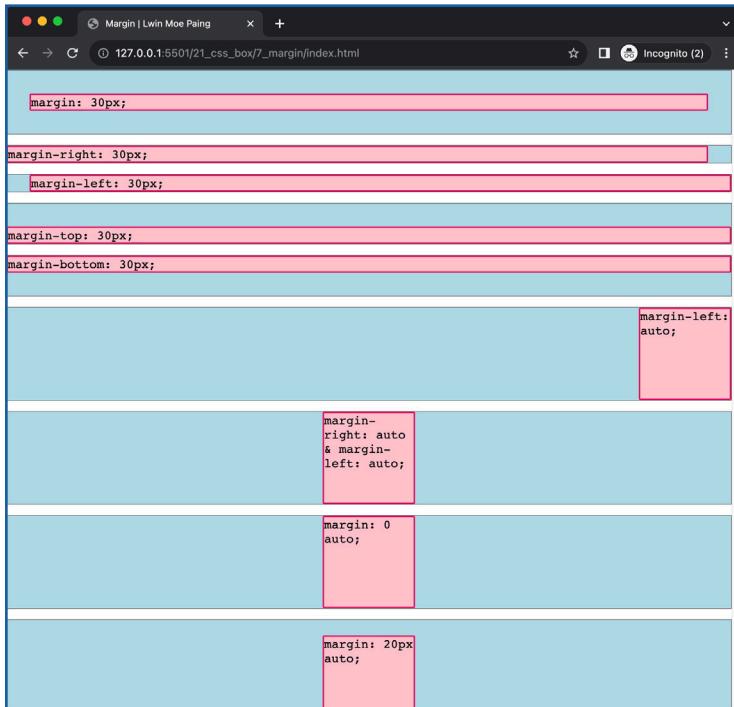
အောက်က style.css အတွက် Gist Code ကို ကူးယူပေးပါ။

<https://gist.github.com/lwinmoepaing/9d3d2b5a605ed15de131ed7bd31332bb>

အရင်ဆုံးအနေနဲ့ CodePen မှာ သွားရောက်စမ်းသပ်ကြည့်နိုင်ပါတယ်။

<https://codepen.io/lwinmoepaing/pen/MWzzGNX>

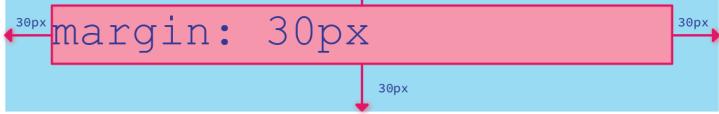
အောက်က ပုံအတိုင်း စာဖတ်သူတို့ Browser တွင်ပေါ်နေပါလိမ့်မည်။



ကျွန်တော်တို့ သတိထားရမှာပါ ဒီအခန်းမှာလေ့လာသွားမယ့် Box တွေအကုန်လုံးက
ပန်းရောင် Box တွေပဲဖြစ်ပါတယ်။ ပေးလိုက်တဲ့ Style Rule တွေကနေတစ်ဆင့် Pink
Box ဘက်က margin အကွာအဝေးတွေကို သတ်မှတ်သွားမှာဖြစ်ပါတယ်။ အောက်ကပုံ
လေးကို တစ်ချက် လေ့လာကြည့်ရအောင် ...။

```
.margin-30px {
    margin: 30px;
}
```

margin: 30px တန်ဖိုးသည် Box ရဲ့ အရပ်
(ငါ) မျက်နှာ စလုံး အပေါ်၊ အောက်၊ ဘယ်၊ ညာ
အကုန်လုံးကို 30px နေရာလွှတ်လုပ်ပေးသွားမှာဖြစ်
ပါတယ်။



```
margin: 30px
```

```
.margin-right-30px {
    margin-right: 30px;
}
```

ရေးသားနည်း margin-right ၏ တန်းစီး 30px သည်
ညာဘက်တစ်ဖက်ကိုသာ နေရာမြေးသွားမှာဖြစ်ပါ
တယ်။



```
margin-right: 30px
```

```
.margin-left-30px {
    margin-left: 30px;
}
```

ရေးသားနည်း margin-left ၏ တန်းစီး 30px သည်
ဘယ်ဘက်တစ်ဖက်ကိုသာ နေရာမြေးသွားမှာဖြစ်ပါ
တယ်။



```
margin-left: 30px
```

```
.margin-top-30px {
    margin-top: 30px;
}
```

ရေးသားနည်း margin-top ၏ တန်းဖိုး 30px သည်
အပေါ်ဘက်တစ်ဖောက်ကိုသာ နေရာခြားသွားမှုပြင်
ပါတယ်။

margin-top: 30px

```
.margin-bottom-30px {
    margin-bottom: 30px;
}
```

ရေးသားနည်း margin-bottom ၏ တန်းဖိုး 30px
သည် အောက်အရာ တစ်ဖောက်ကိုသာ နေရာခြား
သွားမှုပြင်ပါတယ်။

margin-bottom: 30px

```
.margin-left-auto {
    margin-left: auto;
}
```

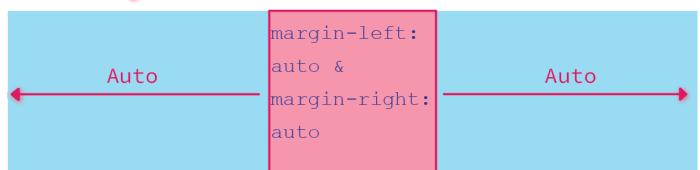
ရေးသားနည်း margin-left ၏ တန်းဖိုး auto သည်
ဘယ်ဘက်မှာရှိသလောက် နေရာရွှေ့တို့ တွက်ချက်
ပြီး ကန်သွားပေးမှုပြင်ပါတယ်။

margin-left:
auto

Auto

```
.margin-left-auto {  
    margin-left: auto;  
}  
.margin-right-auto {  
    margin-right: auto;  
}
```

ရေးသာနည်း margin-left ကော margin-right
ကောတန်ဖိုးကို Auto ပေးလိုက်ပါက နှစ်ဖက်လုံးရဲ့
နေရာလွယ်ကို တွက်ချက်ပြီး ဤားပေးသွားပေးမှာဖြစ်
တဲ့အတွက် အလယ်ကိုရောက်စေချင်တဲ့အပါ အသုံး
ပွားပါတယ်။



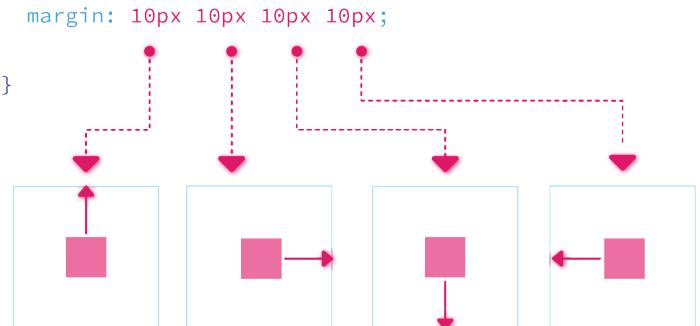
```
.margin-0-auto {  
    margin: 0 auto;  
}
```

margin မှာ တန်ဖိုး (၂) ခြေထဲလည်း ရေးသားလိုရ ပါတယ်။ ပထမတန်ဖိုးကတော့ အပေါ်၏ အောက်ကို ဤားပေးမှာဖြစ်ပြီး ဒု ခုတိယ တန်ဖိုးကတော့ ဘေးနှစ် အက်ရှု၊ တန်ဖိုးအဖြစ်သင်ရောက်မှုဖြစ်ပါတယ်။



.margin-each {

တန်ဖိုး (၄) ခု အသုံးပြုလည်းရှိပါတယ်။ အပေါ်၊
ညာ၊ အောက်၊ ဘယ် အိမိုးတော့ နဲ့မြို့ရေးသားပုံဖြစ်
ပါတယ်။ လက်ရာရင်ပါ အပေါ်ကဗျစတယ်လို့ အလွယ်
ပုံတယ်သားထားနိုင်ပါတယ်။



CSS Padding အကြောင်း

“21_css_box” ဖိုလ်ဒါအတွင်းမှာ ဖိုလ်ဒါအသစ် “8_padding” ဆိုပြီး တည်ဆောက်လိုက်ပါမယ်။ “8_padding” အတွင်းထဲမှာ “index.html” နဲ့ “style.css” ဆိုပြီး HTML နဲ့ CSS ဖိုင် (၂) ခုတည်ဆောက်လိုက်ပါမယ်။

အောက်က index.html အတွက် Gist Code ကို ကူးယူပေးပါ။

<https://gist.github.com/lwinmoepaing/69e3422194107669fea5133df555bef8>

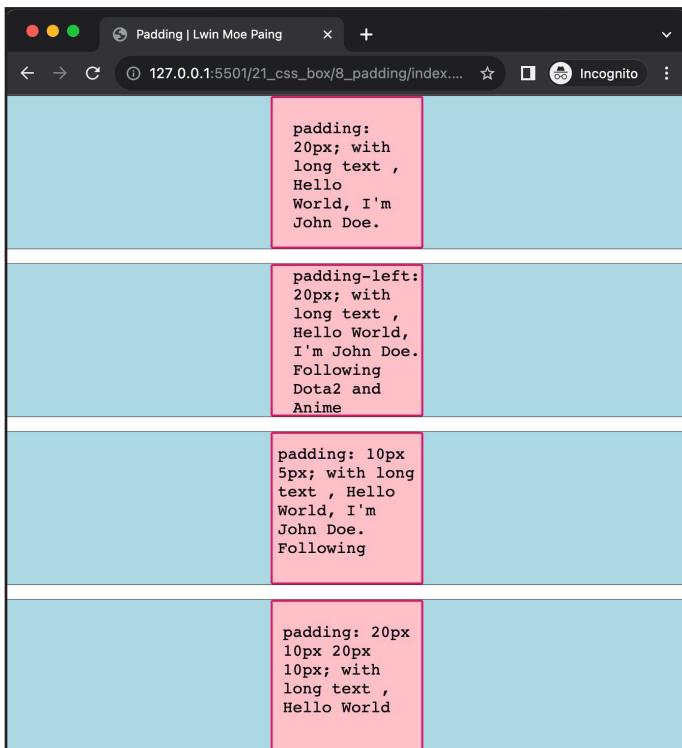
အောက်က style.css အတွက် Gist Code ကို ကူးယူပေးပါ။

<https://gist.github.com/lwinmoepaing/98f06ec5ca6b65fcde551752337ee0de>

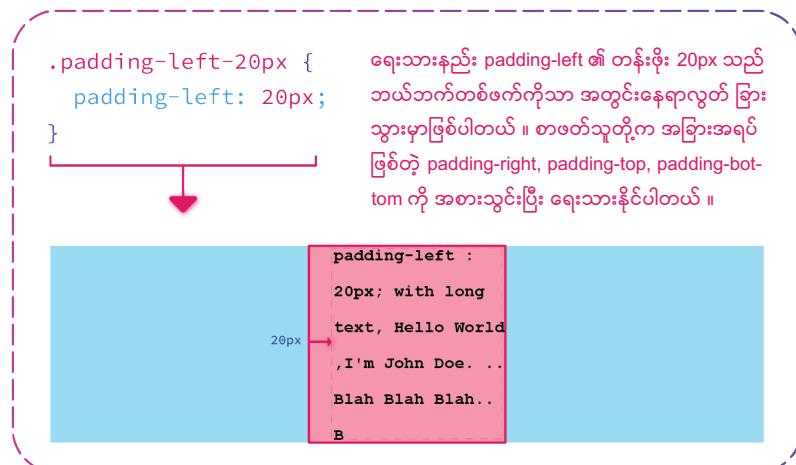
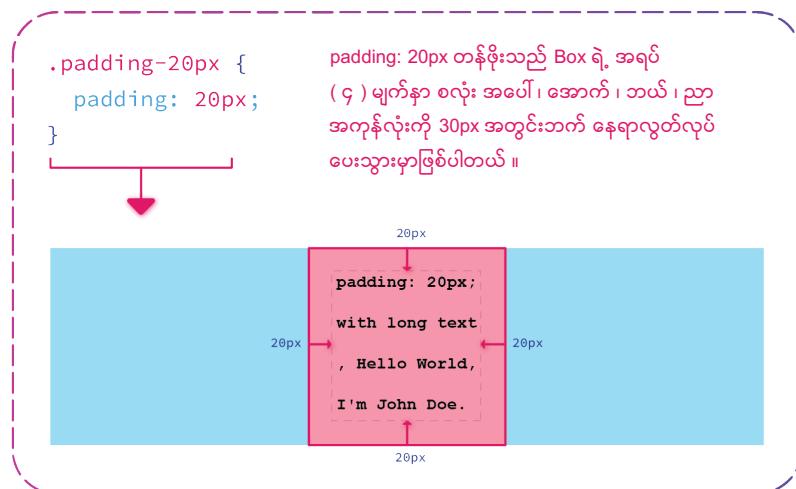
Codepen တွင် အောက်က လင့်ခံမှ တိုက်ရိုက်စမ်းနိုင်ပါသည်။

<https://codepen.io/lwinmoepaing/pen/oNQQVRd>

တဖတ်သူတို့ Browser တွင် အောက်ပါပုံတိုင်းပေါ်လာပါမယ်။



လေ့လာသွားမယ့် Box တွေအကုန်လုံးက ပန်းရောင် Box တွေပဲဖြစ်ပါတယ်။ ပေးလိုက်တဲ့ Style Rule တွေကနေတစ်ဆုံး Pink Box ဘက်က အတွင်းဘက် Padding အကွာအဝေးတွေကို သတ်မှတ်သွားမှုဖြစ်ပါတယ်။ ဒေါက်ကပုံလေးကို တစ်ချက် လေ့လာကြည့်ရအောင် ...။



`.padding-10px-5px {
 padding: 10px 5px;
}`

padding မှာ တန်ဖိုး (၂) ခုဖြင့်လည်း ရေးသားလို့ရ
ပါတယ်။ ပထမတန်ဖိုးကတော့ အပေါ်နဲ့ အောက်ကို
ဤပေးမှုဖြစ်ပြီး။ ဒုတိယ တန်ဖိုးကတော့ ဘေးနှစ်
ကော်၍၊ တန်ဖိုးအပြစ်သက်ရောက်မှာဖြစ်ပါတယ်။

10px
padding: 10px 5px;
with long text,
Hello World
, I'm John Doe. . .
Blah Blah Blah.
10px
5px
5px
10px

`.padding-each {
 padding: 20px 10px 20px 10px;
}`

တန်ဖိုး (၄) ခု အသုံးပြုပုံလည်းရှိ
ပါတယ်။ အပေါ်၊ ညာ၊ အောက်၊
ဘယ် ဆိပ်းတော့ ခွဲပြီးရေးသားပဲ
ဖြစ်ပါတယ်။

အခုခံရင် စာဖတ်သူတို့လည်း padding , margin , border တို့ကို လေ့လာသိရှိပြီးဖော်
ပြီလို့ မှတ်ယူလိုက်ပါမယ့်။ နောက်တစ်မျက်နှာမှာ ကျွန်တော်တို့ Box Shadow ကို
လေ့လာသွားကြပါမယ်။

CSS Box Shadow များအကြောင်း

Box တွေမှာ Shadow အရိပ်ကိုဖော်ပြချင်တဲ့အခါမှာ box-shadow ဆိုတဲ့ Property ကိုအသုံးပြုပြီးရေးသားနိုင်ပါတယ်။ box-shadow ကို တန်ဖိုးပေးတဲ့အခါမှာ Horizontal-offset တန်ဖိုး၊ Vertical-offset တန်ဖိုး၊ Blur Distance (မှန်စိုးစေသော) တန်ဖိုး၊ Spread Shdow (အရိပ်ကို အပေါ်အောက်ဘယ်ညာ အရပ်လေးများနာကို ချွဲကားစေသော တန်ဖိုး) နဲ့ Color (အရိပ်အရောင်) တန်ဖိုးတို့ကိုထည့်ပေးရမှာပါ။

"21_css_box" ဖို့လည်းကောင်းမှာ ဖို့လည်းကောင် "9_box_shadow" ဆိုပြီး တည်ဆောက်လိုက်ပါမယ်။ "9_box_shadow" အတွင်းထဲမှာ "index.html" နဲ့ "style.css" ဆိုပြီး HTML နဲ့ CSS ဖိုင်(၂) စုတည်ဆောက်လိုက်ပါမယ်။

အောက်က index.html အတွက် Gist Code ကို ကူးယူပေးပါ။

<https://gist.github.com/lwinmoepaing/9956b9b76e125904b2fdc04714383c03>

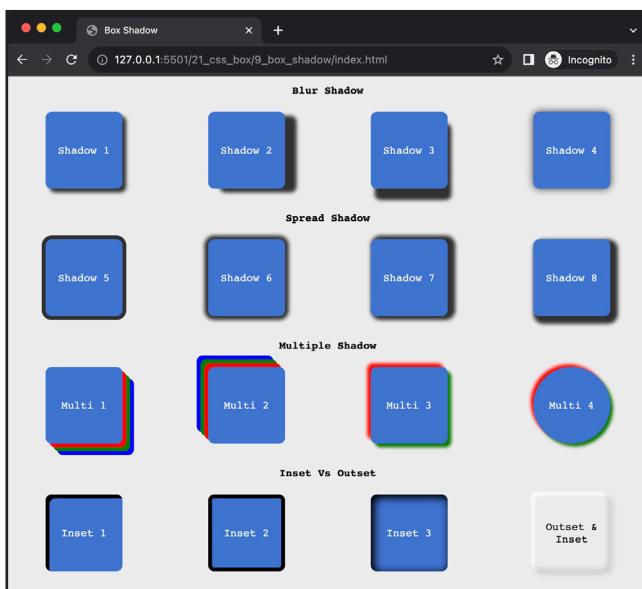
အောက်က style.css အတွက် Gist Code ကို ကူးယူပေးပါ။

<https://gist.github.com/lwinmoepaing/95df43402420ea90d3d6cefbe3427ff>

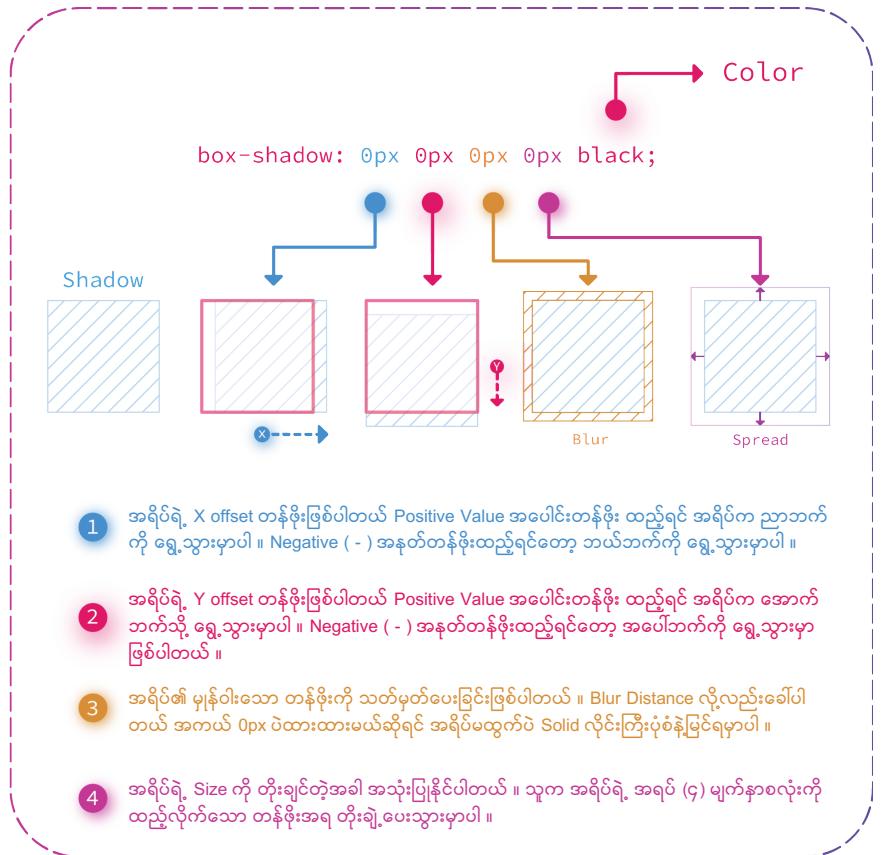
အရင်ဆုံးအနေနဲ့ CodePen မှာ သွားရောက်စမ်းသပ်ကြည့်နိုင်ပါတယ်။

<https://codepen.io/lwinmoepaing/pen/bGQZzLe>

အောက်က ပုံအတိုင်း စာဖတ်သူတို့ Browser တွင်ပေါ်နေပါလိမ့်မည်။



အရင်ဆုံး အောက်က ပုံလေးထဲမှာ Shadow ဆိုတဲ့ အပြာရောင် အစဉ်းကြားလေး
က နောက်ခံ အရိပ်ကို ကိုယ်စားပြုပါတယ်။ ဒါဆိုရင် Value တန်ဖိုး (၅) ခုကို အရင်
လေ့လာကြည့်ရအောင် ...



ကျွန်ုတ်ပေးထားသော Code ကို စာဖတ်သူတို့ Browser ပေါ်တွင် Run ပြီးသွားပြီလို့
သတ်မှတ်လိုက်ပါမယ်။ ဒါဆိုရင် ပထမဆုံး အတန်းလေးကို လေ့လာကြည့်ရအောင် ... ။



Shadow 1 Box ရဲ့ တန်ဖိုး ရေးသားပုံ

```
box-shadow: 5px 5px 5px 0px #333;
```

X-offset: 5px , Y-offset: 5px , Blur - 5px ဖြစ်တဲ့အတွက်
ညာဘက်ကို 5px နဲ့ အောက်ဖက်ကို 5px ယူသွားပြီး မှန်စိုးတန်ဖိုး 5px အနေနဲ့
သတ်မှတ်ပေးသွားတာပဲဖြစ်ပါတယ်။

Shadow 2 Box ရဲ့ တန်ဖိုး ရေးသားပုံ

```
box-shadow: 15px 5px 5px 0px #333;
```

X-offset: 15px , Y-offset: 5px , Blur - 5px ဖြစ်တဲ့အတွက်
ညာဘက်ကို 15px ယူသွားတဲ့အတွက် ပိုပိုသာသာ ညာဘက်မှာ အရိပ်က ပိုရောက်နေ
မှာဖြစ်ပါတယ်။ Y-offset နဲ့ အောက်ဖက်ကို 5px ယူသွားပြီး မှန်စိုးတန်ဖိုး 5px အနေ
နဲ့ သတ်မှတ်ပေးသွားတာပဲဖြစ်ပါတယ်။

Shadow 3 Box ရဲ့ တန်ဖိုး ရေးသားပုံ

```
box-shadow: 5px 15px 5px 0px #333;
```

X-offset: 5px , Y-offset: 15px , Blur - 5px ဖြစ်တဲ့အတွက်
ညာဘက်ကို 5px နဲ့ Y-offset 15px ယူသွားတဲ့အတွက် ပိုပိုသာသာ အောက်ဘက်မှာ
အရိပ်က ပိုရောက်နေမှာဖြစ်ပါတယ်။ Blur မှန်စိုးတန်ဖိုး 5px အနေနဲ့ သတ်မှတ်ပေး
သွားတာပဲဖြစ်ပါတယ်။

Shadow 4 Box ရဲ့ တန်ဖိုး ရေးသားပုံ

```
box-shadow: 0px 0px 10px 0px #333;
```

X-offset: 0px , Y-offset: 0px , Blur - 10px ဖြစ်တဲ့အတွက် Offset
နှစ်ခုလုံးကို (0) တန်ဖိုး သတ်မှတ်ထားတဲ့အတွက် အရိပ်က ဘယ်မှုမရွှေ့တော့ပဲ
အလယ်မှာပဲ တည်ရှိမှာဖြစ်ပြီး။ Blur မှန်စိုးတန်ဖိုး 10px အနေနဲ့ သတ်မှတ်ပေး ထား
တဲ့အတွက် အမှန် အရိပ်ကသာ ဖြန့်ကျယ်ပြီး ထွက်လာမှာဖြစ်ပါတယ်။

စာဖတ်သူတို့က တန်ဖိုးများတို့ ပြောင်းလဲ သတ်မှတ်ပြီး လိုချင်သလို စမ်းသပ်အသုံးပြု
နိုင်ပါတယ်။

ဒုတိယ အတန်းကို လေ့လာကြည့်ရအောင် ... ။



Shadow 5 Box ရဲ့ တန်ဖိုး ရေးသားပုံ

box-shadow: 0px 0px 0px 5px #333;

X-offset: 0px , Y-offset: 0px , Blur - 0px , Spread - 5px

တန်ဖိုး (၃) ခုစလုံး ၀ ဖြစ်နေပြီး **Spread** တန်ဖိုးတစ်ခုတည်းသာ ၅px ပေးထား လိုက်ပါ တယ်။ အဲအတွက် အနောက်က Shadow ရဲ့ Size က ၅px ရှိကြီးနေတာဖြစ်ပြီး **Blur** မှုန်ဝါး တန်ဖိုးလဲမပေးထားတဲ့အတွက် Solid Style နဲ့ အရိပ်မည်းကြီးအဖြစ်မြင်ရတာပါ။

Shadow 6 Box ရဲ့ တန်ဖိုး ရေးသားပုံ

box-shadow: 0px 0px 5px 5px #333;

X-offset: 0px , Y-offset: 0px , Blur - 5px , Spread - 5px

Offset တန်ဖိုး (၂) ခုစလုံး ၀ ဖြစ်နေပြီး **Blur** တန်ဖိုးနှင့် **Spread** တန်ဖိုးသာ ၅px ပေးထား လိုက်ပါတယ်။ Box ရဲ့အနောက်က Size ချယ့်ထားတဲ့ **Spread** ကို **Blur** မှုန်ဝါး လိုက်သလိုဖြစ်သွားမှာပါ။

Shadow 7 Box ရဲ့ တန်ဖိုး ရေးသားပုံ

box-shadow: 5px 0px 5px 5px #333;

X-offset: 5px , Y-offset: 0px , Blur - 5px , Spread - 5px

Shadow 6 Box ရဲ့ တန်ဖိုးကိုမှ **X-offset** တန်ဖိုး **5px** သာထည့်လိုက်တဲ့ အတွက် အရိပ်က ညာဘက်ကို **5px** ရွှေ့သွားမှာဖြစ်ပါတယ်။

Shadow 8 Box ရဲ့ တန်ဖိုး ရေးသားပုံ

box-shadow: 5px 5px 5px 5px #333;

X-offset: 5px , Y-offset: 5px , Blur - 5px , Spread - 5px

Shadow 7 Box ရဲ့ တန်ဖိုးကိုမှ **Y-offset** တန်ဖိုး **5px** သာထည့်လိုက်တဲ့ အတွက် အရိပ်က အောက်ဘက်ကို **5px** ရွှေ့သွားမှာဖြစ်ပါတယ်။

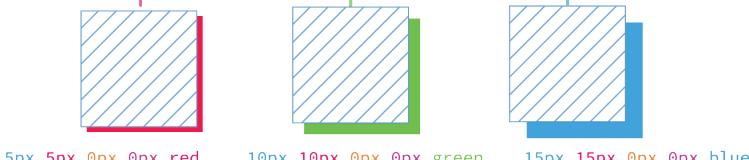
ဒီတစ်ခါ Multiple Shadow ရေးသားနည်းကို လေ့လာကြည့်ရအောင် ... !!

```
box-shadow: 5px 5px 0px 0px red,
            10px 10px 0px 0px green,
            15px 15px 0px 0px blue;
```



BoxShadow အရိပ်များကို တစ်လိုင်းတည်းမြင့် အရိပ် (တစ်) ခုထပ်ပို့ပြီးနှင့်တိုးရေးသားနိုင် ပါသည်။ Red Shadow, Green Shadow နဲ့ Blue Shadow ကို ပေါင်းရေးသားထားမြင်းဖြစ်ပါတယ်။

```
box-shadow: 5px 5px 0px 0px red,
            10px 10px 0px 0px green,
            15px 15px 0px 0px blue;
```



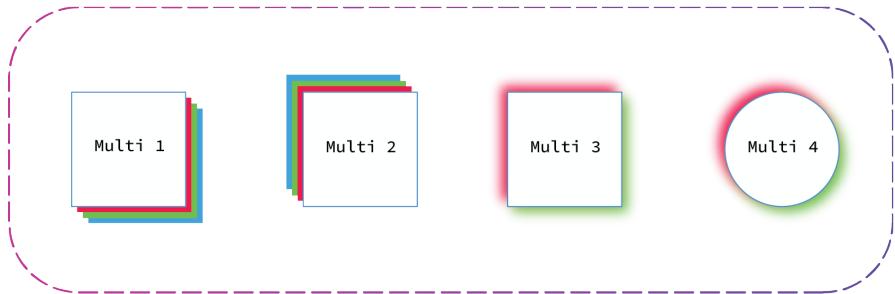
① ပထာဏ္ဍာင်းက အနီးရောင် Shadow ကိုရေးသားထားတော်ဖြစ်ပါတယ်။ X offset - 5px နဲ့ Y offset - 5px ပဲပို့ပြီး Blur မဝါဒ္ဓအတွက် Solid Style ပုံစံဖြစ်နေပြီး ညာဘက်နဲ့ အောက်ဘက်ကို နည်းနည်း ရောက်နေဖူးဖြစ်ပါတယ်။

② ခုတိယလိုင်းက အနီးရောင် Shadow ကိုရေးသားထားတော်ဖြစ်ပါတယ်။ X offset - 10px နဲ့ Y offset - 10px ပဲပို့ပြီး Blur မဝါဒ္ဓအတွက် Solid Style ပုံစံဖြစ်နေပြီး ညာဘက်နဲ့ အောက်ဘက်ကို နည်းနည်း နှင့်ခုတိယလိုင်းနေဖူးဖြစ်ပါတယ်။

③ တတိယလိုင်းက အပြာရောင် Shadow ကိုရေးသားထားတော်ဖြစ်ပါတယ်။ X offset - 15px နဲ့ Y offset - 15px ပဲပို့ပြီး Blur မဝါဒ္ဓအတွက် Solid Style ပုံစံဖြစ်နေပြီး ညာဘက်နဲ့ အောက်ဘက် ပုံပြီးတော့ အမြား အရိပ်များဆင် ရောက်ရှုပို့နေမှာဖြစ်ပါတယ်။

မှတ်ချက် !! !! စတိုင် Rule သတ်မှတ်ရာတွင် နောက်ဆုံး အပိတ်တွင် “ ; ”
SemiColon ကို အသုံးပြုထားသည်ကို သတိပြုရမှာဖြစ်ပြီး အရိပ်များသတ်မှတ် ရာတွင် ထပ်လို့သော အရိပ်များပြီးသွားသောအခါ “ , ” Comma ကော်မာ ကို အသုံးပြုရ မှာဖြစ်ပါတယ်။

တတိယ အတန်းကို လေ့လာကြည့်ရအောင် ... ။



Multi 1 Box ကတေသူ ကျွန်ုပ်တော် Example အနေနဲ့
ထပ်မပြောတော့ပါဘူး ။

Multi 2 Box ရဲ့ တန်ဖိုး ရေးသားပံ့

```
box-shadow: -5px -5px 0px 0px red,
            -10px -10px 0px 0px green,
            -15px -15px 0px 0px blue;
```

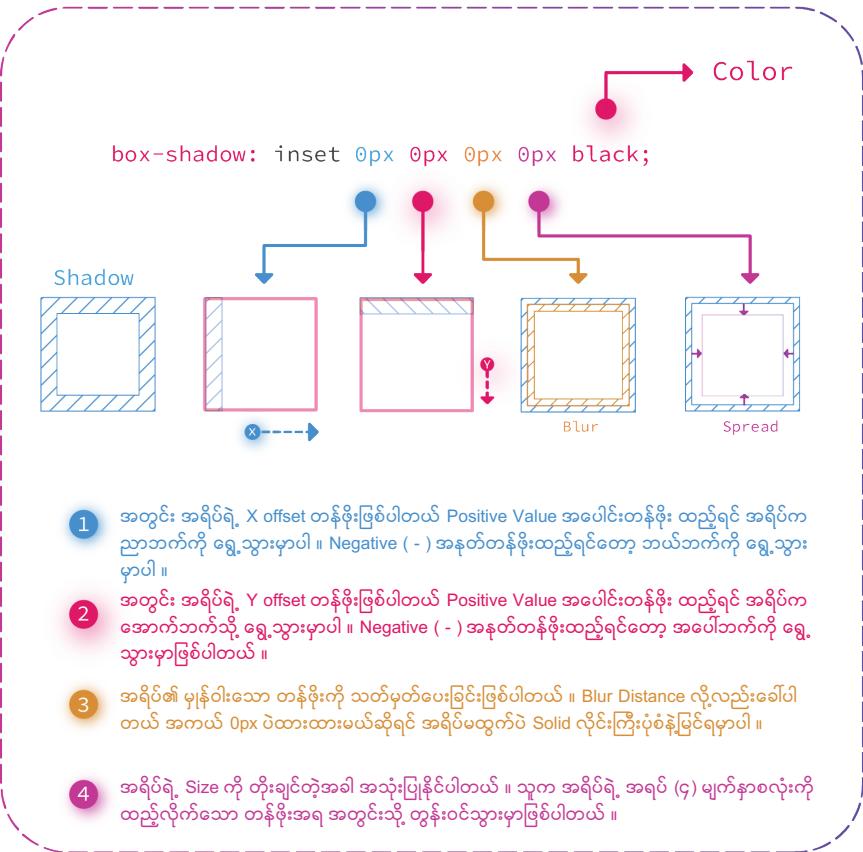
ပထမ အကြောင်းက အနီရောင် Shadow အတွက် X-offset -5px နဲ့ Y-offset -5px ပေးထားတဲ့အတွက် Horizontal အတွက် ပြောင်းပြန် ဘယ်ဘက်ကို တိုးသွားမှပါ ပြီးတော့ Vertical အတွက်အပေါ်ကိုတက်သွားမှာဖြစ်ပါတယ်။ ဒုတိယ အစိမ်းရောင်အတွက်လည်း အတိုင်းပါပဲ အပေါ်ရှည် ဘယ်ဘက်ရယ် -10px အထိတိုးသွားမှပါ။ တတိယ အပြောအတွက်လည်း -15px အထိ ပို့ရောက်သွားမှာပဲဖြစ်ပါတယ်။

Multi 3 Box နဲ့ 4 ရဲ့ တန်ဖိုး ရေးသားပံ့

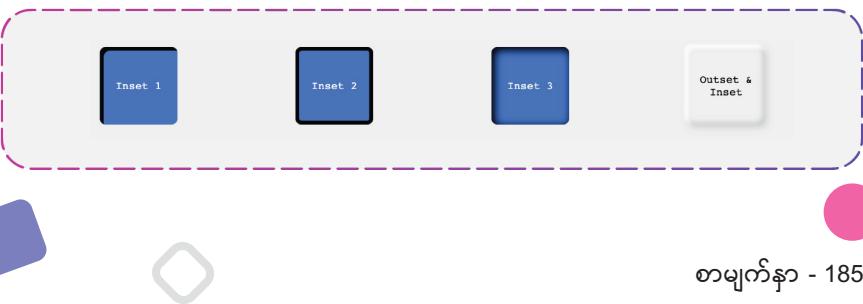
```
box-shadow: -5px -5px 4px 0px red,
            5px 5px 4px 0px green;
```

ပထမ အကြောင်းက အနီရောင် Shadow အတွက် X-offset -5px နဲ့ Y-offset -5px ပေးထားတဲ့အတွက် Horizontal အတွက် ပြောင်းပြန် ဘယ်ဘက်ကို တိုးသွားမှပါ ပြီးတော့ X-offset 5px နဲ့ Y-offset 5px ပေးထားတယ့်အတွက် ညာဘက်နဲ့ အောက်ဘက်ကို ရောက်သွားမှာဖြစ်ပါတယ်။ Blur အတွက် 4px ပေးထားတဲ့အတွက် မျှနှစ်ဝါးတန်ဖိုးနဲ့ နည်းနည်းလေးဝါးနေမှာဖြစ်ပါတယ်။

အတောက လေ့လာခဲ့တယ် Shadow အရိပ်တွေက outset လို့ခေါ်ပါတယ် အပြင်မှာ ပြန့်ထွက်သွားတာဖြစ်ပါတယ်။ အခုပါဘာကတော့ inset ဖြစ်ပါတယ် အတွင်းအရိပ်ကို ပြုလုပ်သွားမှုပါ။ အောက်က ပုံလေးကိုကြည့်ပါ တန်းဖိုး (၆) ခုအနေနဲ့ ဖြစ်သွားမှာဖြစ်ပြီး ထိပ်ဆုံး တန်းဖိုးကို inset ထည့်ပေးလိုက်ရုံပါပဲ တွေးတန်းဖိုးတွေကတော့ လေ့လာရတာခြင်း အတူတူပဲဖြစ်ပါတယ်။



အခုပါနှင့်တော်တို့ inset အတွင်းအရိပ်နဲ့ outset လေ့လာခဲ့ပြီးသော အရိပ်ကွားမြားပုံကို ယုံဉှုံတွဲလေ့လာသွားမှုဖြစ်ပါတယ်။



Inset 1 Box ရဲ့ တန်ဖိုး ရေးသားပုံ

box-shadow: inset 5px 5px 0px 0px black;
X-offset: 5px , Y-offset: 5px , Blur - 0px , Spread - 0px off-set တန်ဖိုးနှစ်ခုလုံးကို 5px သာထည့်လိုက်တဲ့ အတွက် အရိပ်က ညာဘက် နဲ့ အောက် ဘက်ကို 5px ရွှေ့သွားမှာဖြစ်ပါတယ်။ ပြီးတော့ inset အဖြစ်ကြော်ထားတဲ့ အတွက် အတွင်း အရိပ်အဖြစ် ပေါ်သွားပါတယ်။

Inset 2 Box ရဲ့ တန်ဖိုး ရေးသားပုံ

box-shadow: inset 0px 0px 0px 5px black;
Spread - 5px သာ ပါဝင်တဲ့ အတွက် အရိပ်က အရပ်လေးမျက်နှာကနေ အတွင်းသို့ ချဲ ဝင်သွားမှာဖြစ်ပါတယ်။

Inset 3 Box ရဲ့ တန်ဖိုး ရေးသားပုံ

box-shadow: inset 3px 3px 10px 2px black;
X-offset: 3px , Y-offset: 3px , Blur - 10px , Spread - 2px offset တန်ဖိုး(2) ခုက 5px ဖြစ်ပြီး Blur မှန်စိုးတန်ဖိုး 10px နဲ့ Shadow spread size အကျယ်က 2px သတ်မှတ်ထားတာဖြစ်ပါတယ်။ အဲဒေါက်ဘက် နဲ့ ညာဘက် တစ်ဖက်တစ်ချက်ဆိုက အနည်းငယ် Shadow အရိပ်လေးကို တွေ့မြင်ရမှာပါ။

Outset & Inset Box တန်ဖိုး ရေးသားပုံ

**box-shadow: inset 2px 2px 4px 0px #ffffff ,
 inset -2px -2px 8px 0px #c3c3c3 ,
 -3px -3px 4px 0px #ffffff ,
 2px 3px 10px 0px #cacaca ;**

- ပထမ လိုင်းက inset ဖြစ်တဲ့ အတွက် အတွင်းအရိပ်ကိုသတ်မှတ်တာပါ အတွင်းအပေါ်နဲ့ အတွင်းရဲ့ ဘယ်ဘက်ကို အဖြူရောင် အရိပ်လေးပေါ်စေချင်လိုပါ။
- ဒုတိယ လိုင်းကလည်း inset ဖြစ်တဲ့ အတွက် အတွင်းဘက်ပါပဲ သူကအတွင်းရဲ့ အောက် ဘက်နဲ့ ညာဘက်ကို အရိပ်လေးဖြစ်စေချင်တဲ့ အတွက် -2px အနေနဲ့ offsets နှစ်ခုလုံးကိုပေးပြီး ရေးသားခြင်းဖြစ်ပါတယ်။
- တတိယ လိုင်းကတော့ အဖြူရောင် အရိပ်ကို အပြင်ဘက်အပေါ်နဲ့ ဘယ်ဘက်မှာပေါ်စေချင် တယ် အတွက် offsets (J) ခုကို -3px ပေးကာ ရေးသားလိုက်ခြင်းဖြစ်ပါတယ်။
- စတုတ္ထလိုင်းကတော့ အောက်ဘက်နဲ့ ညာဘက်ထောင့်ကို အရိပ်အနက်ရောင်ပေါ်စေချင်တဲ့ အတွက် ရေးသားခြင်းဖြစ်ပါတယ်။

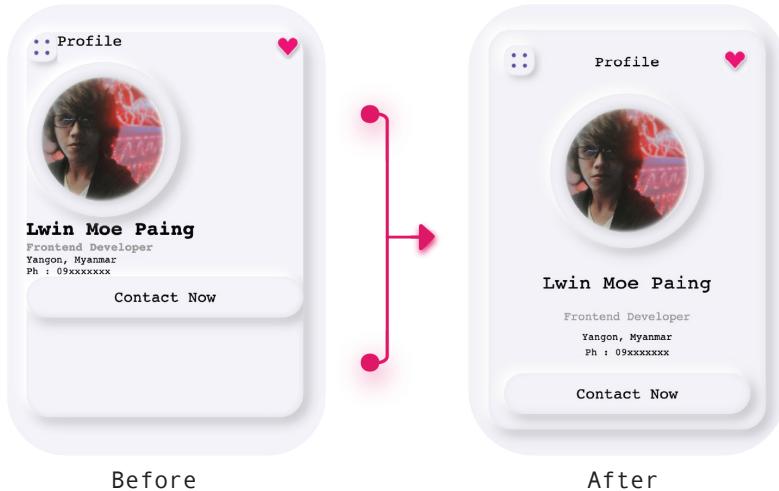
မှတ်ချက်။ ချက်ချင်းတော့ အကုန်နားလည်သွားမှာမဟုတ်ပါဘူး inset နဲ့ outset ကို ရေးအခါး အခါး ဖောင်းကြရေးသားပုံမျိုးရှိတာကို သိစေချင်လို့ ထည့်သွင်းဖော်ပြုခဲ့ပါတယ်။

From Theory to Practice: CSS Boxes

အခုဆိုရင် Practice လုပ်ဖို့အချိန်ရောက်လာပြန်ပါပီ။

CSS Box တွေနဲ့ပါတ်သတ်ပြီး စာဖတ်သူတို့အနေနဲ့ ဘယ်လောက်ထိ နားလည်သွားလဲဆိတာ စမ်းသပ်ကြရမှာဖြစ်ပါတယ်။ အောက်က ပုံလေးကို တစ်ချက်ကြည့်လိုက်ပါ ဘယ်ဘက်က ပုံက ကျွန်တော်တို့ Margin, Padding တွေမလုပ်ရသေးတဲ့အတွက် ကပ်ပြီး ကြည့်လို့မကောင်းဖြစ်နေမှာပါ။

ဒါကို စာဖတ်သူတို့က ညာဘက်ကအတိုင်း ဖြစ်သွားအောင်လို့ CSS Rule ထွေ သတ်မှတ်ပြီးတော့ ပြင်ပေးရမှာဖြစ်ပါတယ်။



အရင်ဆုံးအနေနဲ့ CodePen မှာ သွားရောက်စမ်းသပ်ကြည့်နိုင်ပါတယ်။

<https://codepen.io/lwinmoepaing/pen/jOQRoBY>

အောက်က `index.html` အတွက် Gist Code ကို ကူးယူပေးပါ။

<https://gist.github.com/lwinmoepaing/3dc3b1b1e7c30ec88da0befdad015d20>

အောက်က `predefined.css` ဆိုပြီး ဖိုင်ဆောက်ပြီး အောက်က Gist Code ကို ကူးယူပေးပါ။

<https://gist.github.com/lwinmoepaing/7624f61b3c79ed20f6293c40623b6e6e>

အောက်က `style.css` ဆိုပြီး ဖိုင်ဆောက်ပြီး အောက်က Gist Code ကို ကူးယူပေးပါ။

<https://gist.github.com/lwinmoepaing/c7be6796efe91e8b344a92b7222fd36a>

style.css အတွင်းမှာ စာဖတ်သူတို့က အိမ်စာ Practice ကိုရေးသွင်းသတ်မှတ်ပေးရမှာ ပဲဖော်ပါတယ်။ အောက်ပါအတိုင်း Class Selector များအတွင်းမှာ Margin, Padding, Text Alignment တို့ကို ဖြည့်သွင်းပြီးတော့ ပုံကိုအမှန်တိုင်းဖြစ်အောင် ပြုပြင်ပေးပါ ခင်ဗျာ။



```
21_css_box/10_practice/style.css

.card {
}

.app-bar {
}

.profile-title {
}

.profile-image {
}

.profile-name {
}

.profile-position {
}

.profile-detail {
}

.app-button {
}
```

အခုတော့ အခြေဖော်လိုက်ခင် ကြိမ်းသေ စာဖတ်သူတို့ကိုယ်တိုင် CSS rules များရေးသား နိုင်လိမ့်မယ်လို့ ယုံကြည်ပါတယ်။

ကိုယ်ပိုင်ပုံစံဖြင့် Result အရ ကောင်းမွန်စွာထွက်နေပြီး ဖြစ်သော်လည်း ကျွန်တော်ပေး ထားသော Code နဲ့တူညီမှ မှန်တယ်လို့မဆိုလိုပါ။

style.css ထဲက Code ကို အောက်က Gist မှာသွားရောက် တိုက်ကြည်ပြီးလေ့လာနိုင်ပါ တယ်ၢုံး။

<https://gist.github.com/lwinmoepaing/57744f9d1de284abfe1854baee5e49c9>

အရင်ဆုံးအနေနဲ့ CodePen မှာ သွားရောက်စမ်းသပ်ကြည့်နိုင်ပါတယ်။

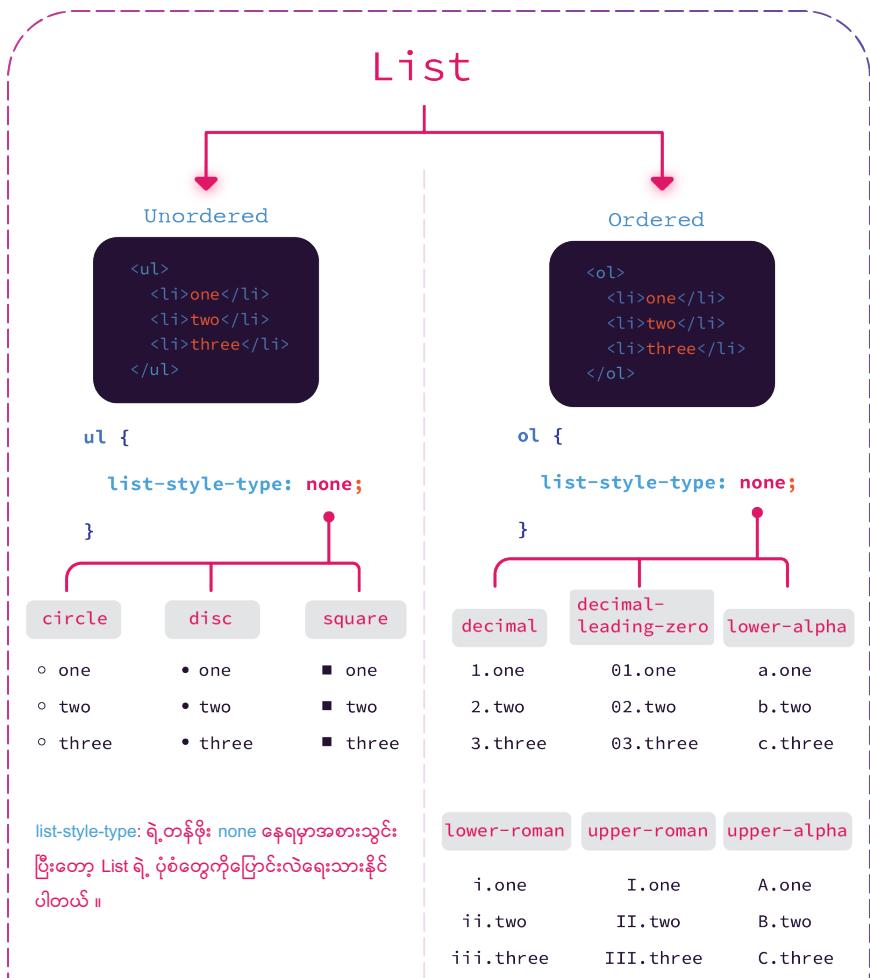
<https://codepen.io/lwinmoepaing/pen/zYMXYKe>

CSS List, Table များအကြောင်း

Browser ပေါ်တွင် List များကို ဖော်ပြရတွင် နှင့် ဆိုတဲ့ Element တွေ အသံးပြုထာကို HTML Elements များ အခန်းမှာ လေ့လာကြပြီးဖြစ်ပါတယ်။ ဤော်တော် Form Controls နဲ့ Table အတွက် <table> <tr> <td> တို့လည်း လေ့လာပြီးသားဖြစ်ပါတယ်။

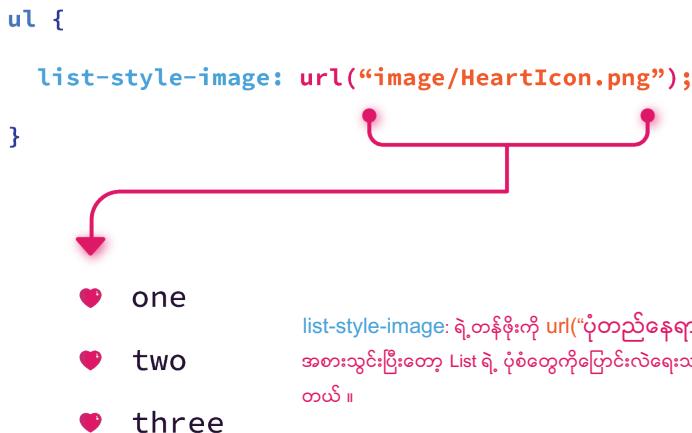
CSS List များအကြောင်း

list-style-type အကြောင်းကိုအရင်ဆုံးလေ့လာကြည့်ရအောင် ...။

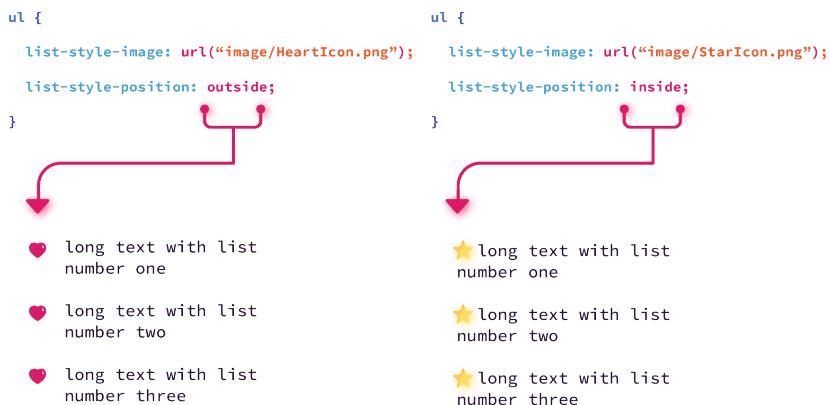


ကျွန်တော်တို့ List ရဲ့ အရှေ့မှာပါတဲ့ အစက်လေးတွေ သို့မဟုတ် နံပါတ်စဉ်တွေကို ငြောင်းလဲသတ်မှတ်တာကိုလေ့လာပြီးဖြစ်ပါတယ်။ ဒီတခါတော့ ကိုယ်ရဲ့

Custom Image (စိတ်ကြိုက်ပုံ) တွေထည့်ပြီး List Style ကိုသတ်မှတ်တာကိုလေ့လာ ကြုံမပယ်ပျော်။

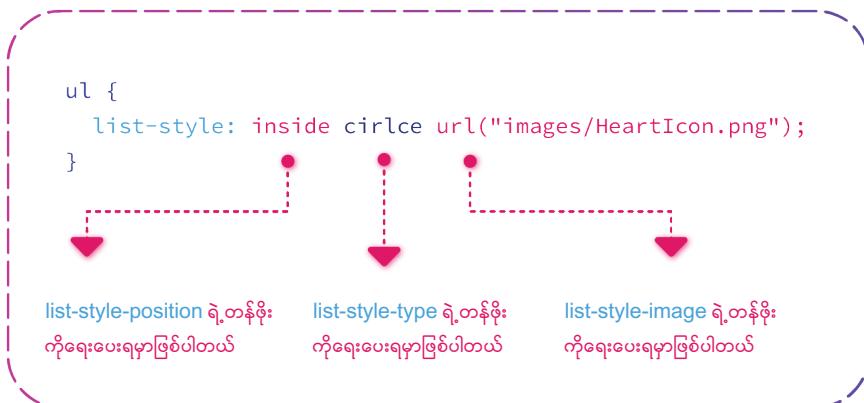


Custom Image (စိတ်ကြိုက်ပုံ) တွေထည့်နည်းကို လေ့လာပြီးတဲ့အခါ စာဖတ်သူတို့က List တွေရဲ့ Position ကိုအောက်ကပုံထဲက အတိုင်းလေ့လာကြရအောင်။



List Position မှာ (J) ပိုင်းပါပါတယ် မူရင်းတန်ဖိုးကတော့ outside ပါ စာတန်းတွေရည် သွားတဲ့ အခါ စာသားက အောက်တစ်လိုင်းကျသွားပါတယ် သို့ပေါ်မယ် ဘေးက List အမှတ်စဉ် (သို့မဟုတ်) ကိုယ်တွေ သတ်မှတ်ထားတဲ့ (စိတ်ကြိုက်ပုံ) ရဲ့ ညာဘက်မှာပဲ စာသားတွေကရှိနေမှာပါ။ ဒါမှာဟုတ် outside ကိုသုံးပြီတော့ နောက်တစ်လိုင်း ဆင်းသွားတဲ့အပါမှာ ဘေးက အမှတ်စဉ်ကို စွာမသွားပဲ inside ဆိုတဲ့အတိုင်း အတွင်းဘက်ကို ရောက်သွားမှာဖြစ်ပါတယ်။

list-style အတိုကောက်ရေးသားနည်းဖြင့် အောက်ပါပုံအတိုင်းရေးသားနိုင်ပါတယ်။



List Style အကြောင်းကို စာဖတ်သူတို့ လေ့လာခဲ့ပြီးဖြစ်ပါတယ်။

ခဲ့တော့ Coding ပိုင်းအရစမ်းလို့ရအောင် စိုလ်ဒါအသစ် “22_css_list” ဆိုပြီး တည်ဆောက်လိုက်ပါမယ်။ “22_css_list” ဆိုတဲ့ စိုလ်ဒါ အတွင်းထဲမှာ “index.html” နဲ့ “style.css” ဆိုပြီး HTML နဲ့ CSS ဖိုင် (J) ခုတည်ဆောက်လိုက်ပါမယ်။

စာဖတ်သူတို့က index.html တွင် ဒီအောက်က Code ကိုကူးယူပြီးစမ်းကြည့်လို့ရပါမယ်။

<https://gist.github.com/lwinmoepaing/d4463b0e3293ebb48AAF4bd9e53bb38d>

style.css တွင် ဒီအောက်က Code ကိုကူးယူပြီးစမ်းကြည့်ပါ။

<https://gist.github.com/lwinmoepaing/a784f6e606a5c47e5785e8c3d077597f>

Codepen မှာလည်းတိုက်ရှိက် စမ်းကြည့်နိုင်ပါတယ်။

<https://codepen.io/lwinmoepaing/pen/rNQgZNR>

CSS Table များအကြောင်း

Table Element များအတွက် စာဖတ်သူတို့ Table လေားများတည်ဆောက်ခြင်း အခန်းမှာ လေ့လာခဲ့ပြီးသားဖြစ်ပါတယ်။ ဒါကြောင် နည်းနည်းမေ့တော့တော့ဖြစ်နေရင် ဒဲ အခန်းကို ပြန်ပြီး လေ့လာဖော်ဖြစ်ဖို့ သတိပေးလိုပါတယ်။ ဟုတ်ပြီ ဒါဆိုကျွန်တော်တို့ အတူတူ ဥပမာ Table လေးတစ်ခုအရင်တည်ဆောက်လိုက်ရအောင်။

HTML

```
<table>
  <tr>
    <th>Name</th>
    <th>Hobby</th>
    <th>Address</th>
  </tr>

  <tr>
    <td>Lwin Moe Paing</td>
    <td>Gaming, Reading</td>
    <td>Yangon</td>
  </tr>

  <tr>
    <td>Kyaw Gyi</td>
    <td>Soccer, Music</td>
    <td>Mandalay</td>
  </tr>

  <tr>
    <td>Aung Aung</td>
    <td>Reading</td>
    <td>Ayeyarwady</td>
  </tr>
</table>
```

CSS

```
table {
  min-width: 600px;
}

table, td, th {
  border: 1px solid #aaaaaa;
}
```



| Name | Hobby | Address |
|----------------|-----------------|------------|
| Lwin Moe Paing | Gaming, Reading | Yangon |
| Kyaw Gyi | Soccer, Music | Mandalay |
| Aung Aung | Reading | Ayeyarwady |

အခုလို HTML အတွင်းမှာ <table> element တစ်ခုတည်ဆောက်ပြီး CSS အတွင်းမှာ Border 1px ဆီ table, th, td အစဉ်တဲ့ elements များကို သတ်မှတ်ပေးလိုက်ပါမယ်။ ဒါ ဆိုရင်ပုံထဲကအတိုင်း Table လေးတစ်ခု Browser ပေါ်မှာပေါ်လာမှာဖြစ်ပါတယ်။ အရင်အခန်းမှာတုန်းက border attribute နဲ့ ရေးသားခဲ့တာကို မှတ်မီးမယ်ထင်ပါတယ်။ အခု အခါမှာတော့ CSS ဖိုင်အတွင်းမှာပဲ border ကိုရေးသားသတ်မှတ်ထားတာ ဖြစ်ပါတယ်။ Default မူလတန်ဖို့ အရ Border Collpase ကြား Spacing အကျာအဝေးက ခြားနားနေမှာဖြစ်ပါတယ် ဒါကို အောက်ကပုံအတိုင်း က်သွားအောင်ရေးသားနိုင်ပါတယ်။

CSS

```
table {
  min-width: 600px;
  border-collapse: collapse;
}

table, td, th {
  border: 1px solid #aaaaaa;
}
```

```
border-collapse: collapse;
```

| Name | Hobby | Address |
|----------------|-----------------|------------|
| Lwin Moe Paing | Gaming, Reading | Yangon |
| Kyaw Gyi | Soccer, Music | Mandalay |
| Aung Aung | Reading | Ayeyarwady |

border-collapse ကို collapse တန်ဖိုးပေးပြီးတော့ ကြားထဲမှာရှိတယ့် Spacing အကွာ အဝေးကို မရှိပဲ ကပ်သွားအောင်ပြလုပ်နိုင်ပါတယ်။ ဒါဆိုရင် Spacing အကွာအဝေး ကို ကိုယ်တိုင် Customize အကွာအဝေးထားချင်တဲ့အခါမှာ အောက်ကန်ည်းအတိုင်း ရေးသားနိုင်ပါတယ်။

CSS

```
table {
    min-width: 600px;
    border-collapse: separate;
    border-spacing: 20px 20px;
}

table, td, th {
    border: 1px solid #aaaaaa;
}
```

```
border-spacing: 20px 20px;
```

| Name | Hobby | Address |
|----------------|-----------------|------------|
| Lwin Moe Paing | Gaming, Reading | Yangon |
| Kyaw Gyi | Soccer, Music | Mandalay |
| Aung Aung | Reading | Ayeyarwady |

border-collapse ကို separate တန်ဖိုးပေးပြီးတော့ ကြားထဲမှာရှိတယ့် Spacing အကွာ အဝေးကို border-spacing နဲ့ ကြောင်းလဲနိုင်ပါတယ်။ သို့ပေါ်ယူ border-collapse ကို sperate က မကြော်ပေးလည်းရပါတယ် ဘာလို့ဆို နိုင်မှုရင်းတန်ဖိုးအဖြစ် Browser က သတ်မှတ်ထားလိုပါ။ border-spacing ပထမတန်ဖိုး 20px သည် Horizontal အလျားလိုက် အကျယ်အကွာအဝေးဖြစ်ပြီးတော့ ဒုတိယ 20px ကတော့ Vertical ထောင်လိုက် အကျယ်အကွာအဝေးဖြစ်ပါတယ်။ စာဖတ်သူတို့က နှစ်သက်ရာတန်ဖိုး များအတားသွင်းပြီး စမ်းသပ်ရေးသားနိုင်ပါတယ်။

CSS

```
table {
    min-width: 600px;
    border-collapse: collapse;
    border: 1px solid #fdfdfd;
}

td, th {
    padding: 10px 10px;
}

table tr:nth-child(even) {
    background-color: #fdfdfd;
}
```

```
nth-child(even)
```

| Name | Hobby | Address |
|----------------|-----------------|------------|
| Lwin Moe Paing | Gaming, Reading | Yangon |
| Kyaw Gyi | Soccer, Music | Mandalay |
| Aung Aung | Reading | Ayeyarwady |

ဒီတစ်ခုကတော့ pseudo class တစ်ခုဖြစ်တဲ့ nth-child ကိုအသုံးပြုထားတာပါ။ ဒီလို selector ကိုသုံးပြုပြီးတော့ စုတိန်းနံပါတ်လိုက် (ဥပမာ- ၂၊ ၄၊ ၆) လိုင်းကြောက်တွေကို သာ Background Color ကိုပေးချင်အခါ အသုံးပြုပါတယ် (သို့မဟုတ်) မကိန်းနံပါတ် တွေပဲ ပေးချင်ရင်တော့ tr:nth-child(odd) ဆိုပြီးကြောင်းလည်းအသုံးပြုလို့ရပါတယ်။

Table Style အကြောင်းကို စာဖတ်သူတို့ လေ့လာခဲ့ပြီးဖြစ်ပါတယ်။

အဲတော့ Coding ပိုင်းအရမ်းလို့ရအောင် ဖိုလ်ဒါအသစ် “23_css_table” ဆိုပြီး
တည်ဆောက်လိုက်ပါမယ်။ “23_css_table” ဆိုတဲ့ ဖိုလ်ဒါ အတွင်းထဲမှာ “index.html”
နဲ့ “style.css” ဆိုပြီး HTML နဲ့ CSS ဖိုင် (J) ရတည်ဆောက်လိုက်ပါမယ်။

စာဖတ်သူတို့က index.html တွင် ဒီအောက်က Code ကိုကူးယူပြီးစမ်းကြည့်လို့ရပါမယ်။

<https://gist.github.com/lwinmoepaing/142bc8545ca633e3bcf260a11edcdd3b>

style.css တွင် ဒီအောက်က Code ကိုကူးယူပြီးစမ်းကြည့်ပါ။

<https://gist.github.com/lwinmoepaing/3fd2267a9a81cad382888e68a5f09ed8>

Codepen မှာလည်းတိုက်ရိုက် စမ်းကြည့်နိုင်ပါတယ်။

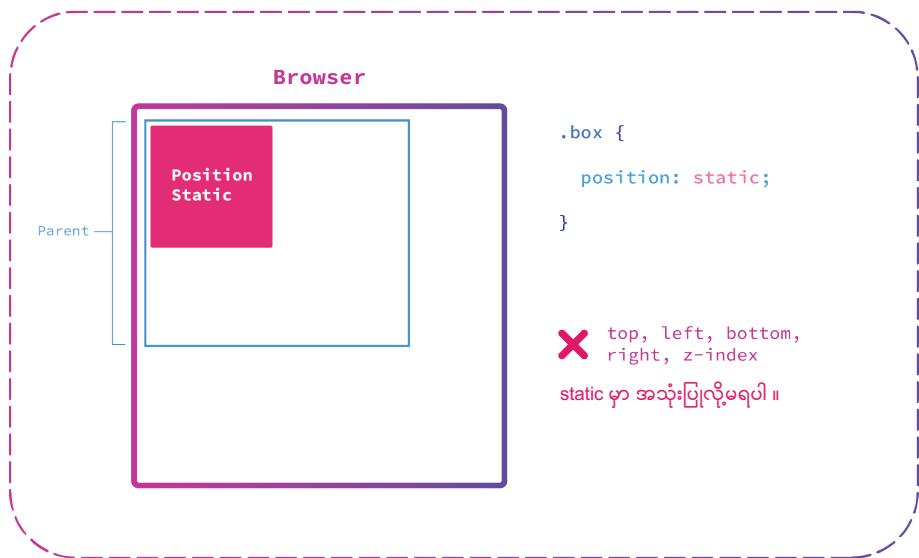
<https://codepen.io/lwinmoepaing/pen/eYQaxPW>

CSS Position, Transition, Background Image

CSS Positions များအကြောင်း

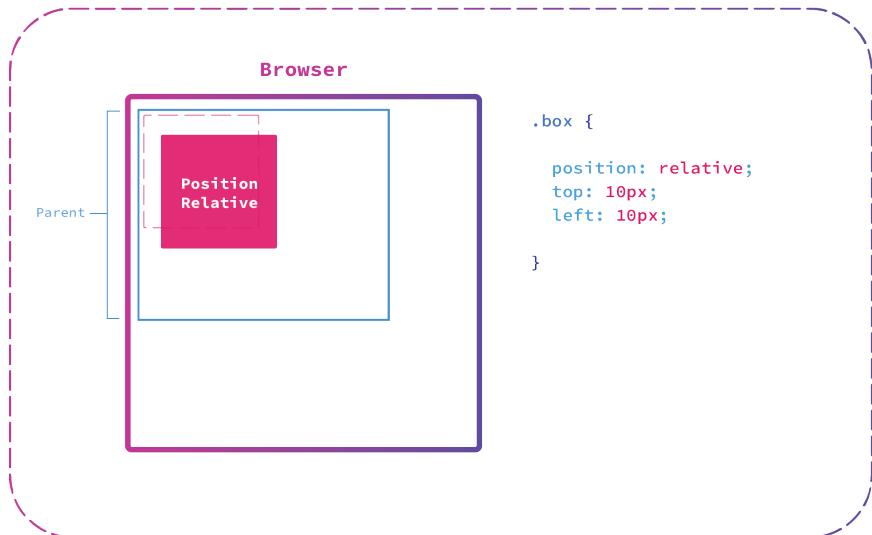
တာဖတ်သူတို့ Boxex အခန်းမှာတုန်းက Padding ၊ Margin တို့ကိုလေ့လာခဲ့ပြီးတဲ့အခါ မှာ ပုံလေးမှာပါတဲ့ Top, Left, Bottom, Right ဆိုပြီး သတိထားမိခဲ့ပါသလား ? ဟုတ်ပါ ပြီ ဒါဆိုရင် ဒီအခန်းမှာ Box တွေရဲ့ Position တွေကို ကျွန်ုတ်တော်တို့ သတ်မှတ်ပါမယ်။

Position ရဲ့တန်ဖိုးတွေမှာ static , absolute, fixed, sticky ဆိုပြီး သုံးလေ့ရှိပါတယ်။ အရင်ဆုံး မူလတန်ဖိုးဖြစ်တဲ့ static ကိုလေ့လာကြည့်ရအောင်။

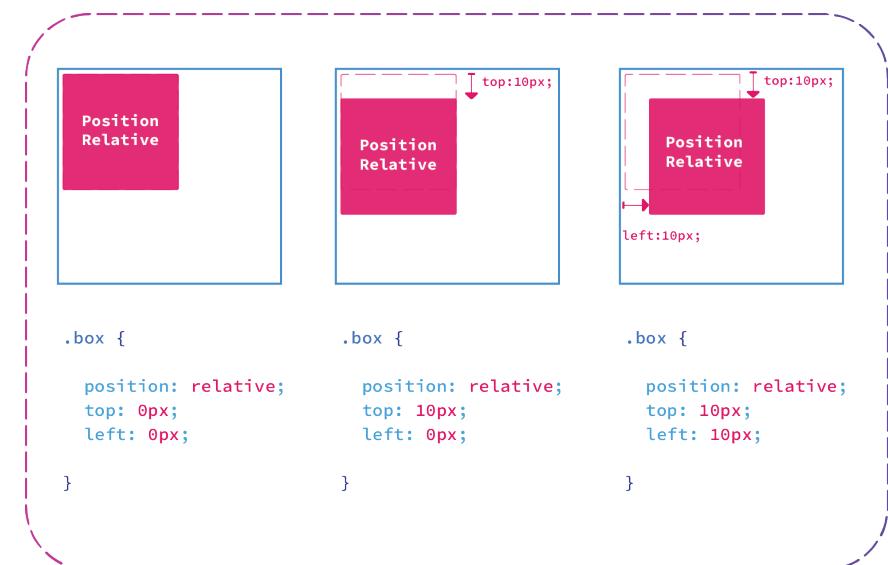


Static က မူလတန်ဖိုးဖြစ်ပါတယ်။ နောက် တစ်မျက်နှာမှာ ကျွန်ုတ်တော်တို့ position: relative ကိုလေ့လာသွားကြပါမယ်။

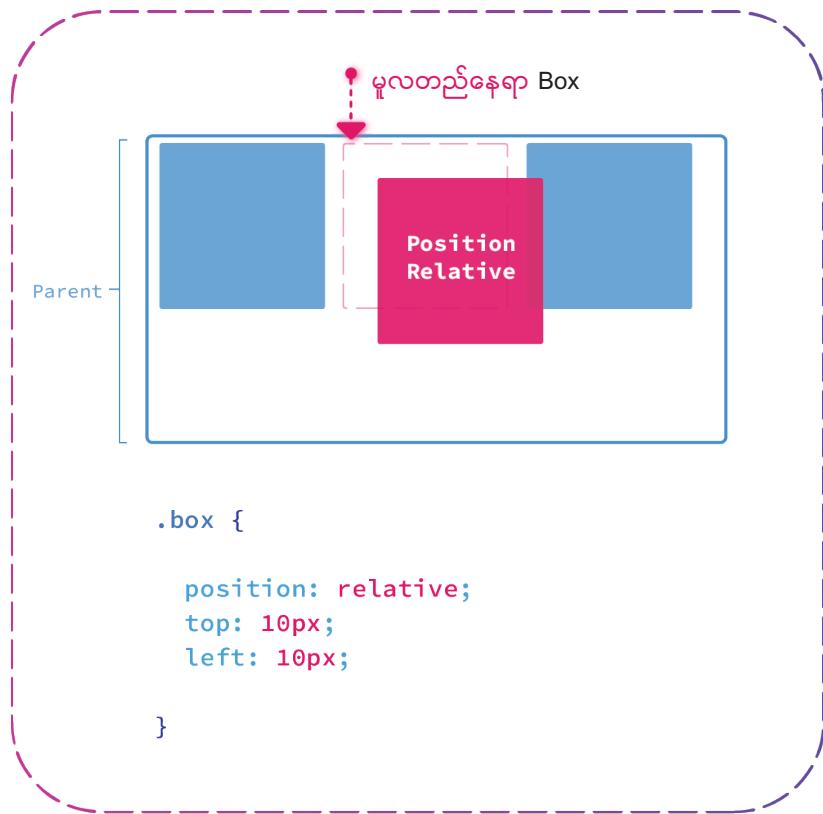
position: relative အတွက် အောက်ကပုံလေးကိုတစ်ချက်ကြည့်ကြည့်ပါ။



အရင်ဆုံး အနိုင်ရောင် Box က သူရဲ့ Parent အပြာရောင် Box နဲ့ အနည်းငယ်ကွာဟန် တာကိုမြင်ရပါလိမ့်မယ်။ အထဲတွင်ရေးထားတဲ့ top: 10px နဲ့ left: 10px ကြောင့် အကွာအဝေး ဤော်သွားတာဖြစ်ပါတယ်။ ဒါကို အောက်ကပုံကိုကြည့်ပြီး ရှင်းလင်းပေးထားပါတယ်။

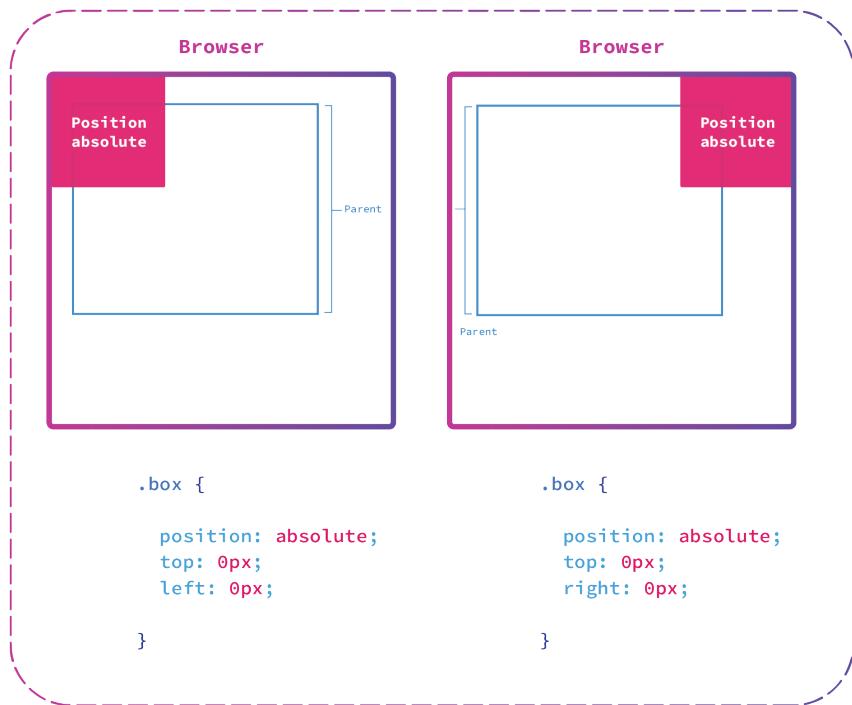


ဒုတိယ Box အရ `top: 10px` ကိုပေးလိုက်တဲ့အခါ အပေါ်ကနေ `10px` ကွာသွားတာ
ဖြစ်ပါတယ်။ တတိယမှာတော့ `left: 10px` ကို ထပ်ပေါင်းထည့်လိုက်တာကြော
င့် ဘယ်ဘက်ကနေစပီး `10px` အကွာအဝေး ဗျားသွားတာဖြစ်ပါတယ်။ သတိပြုရမှာ
က အနောက်က အစက်အပြောက် ဖြင့် ဆွဲထားသော Box က သူရဲ့နှစ်မှုရင်းရှိရမယ့်
တည်နေရာကိုကိုယ်စားပြုခြင်းဖြစ်ပါတယ်။ relative တန်ဖိုးအတွက် ထပ်ပြီးလေ့လာ
ကြည့်ရအောင်



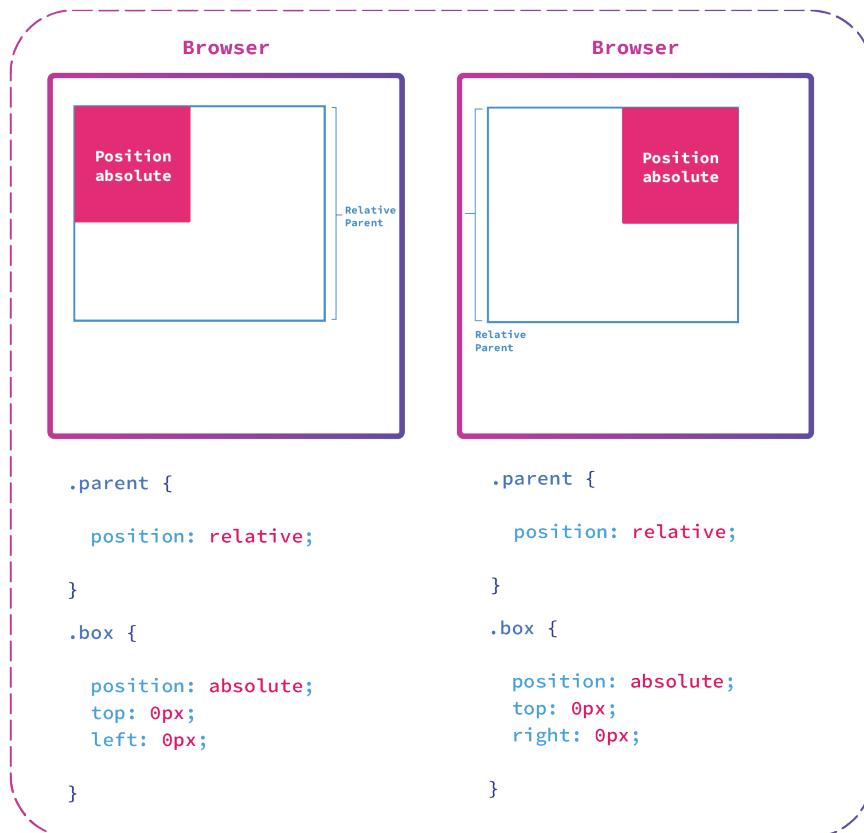
အနီးရောင် Box ကိုပဲ `top: 10px` နဲ့ `left: 10px` ကိုထည့်ထားတာကြောင့် သူရဲ့
လက်ရှိတည်နေရာကနေ ညာဘက်နဲ့ အောက်ဘက်ကို အနည်းငယ်တိုးသွားမှုဖြစ်ပါ
တယ်။ ထူးဗျားချက်အရ အပြောရောင် Box တွေက သူရဲ့ အရင်ကတည်ရှိရမယ့် နေရာ
ကို အစားထိုး ဝင်ရောက်နိုင်စွမ်းမရှိပါဘူး။ `position: relative` အသုံးပြုခဲ့ရင် သူရဲ့
တည်ရှိခဲ့တဲ့ အရင်မှုလ နေရာက အစက်အပြောက် Box လေးအတိုင်း နေရာယူသွားမှာ
ပဲဖြစ်ပါတယ်။

position: absolute အတွက် လေ့လာကြည့်ရအောင် ... !!



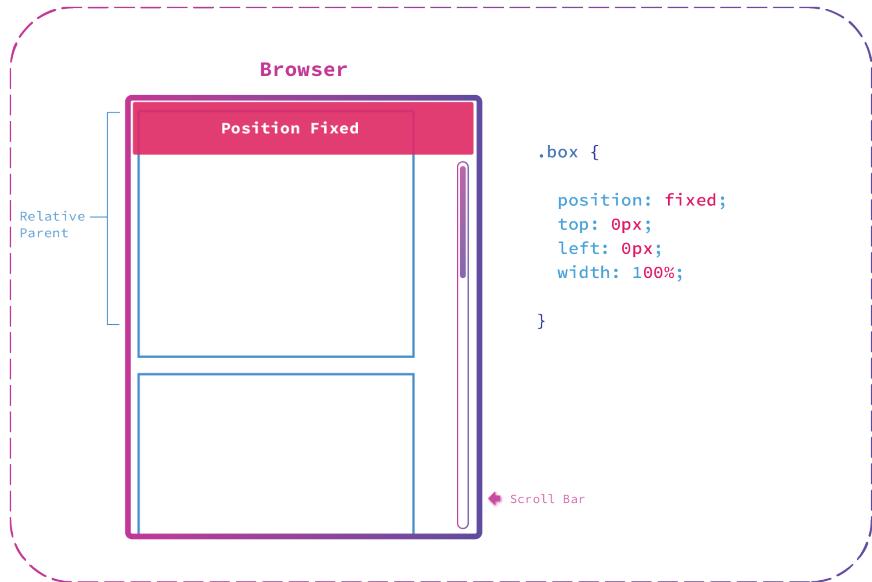
ဘယ်ဘက်က **absolute Box** အတွက် **top: 0px** နဲ့ **left: 0px** ပေးလိုက်တဲ့အခါ Browser Box အထိကိုရောက်သွားပြီး ကိုယ့်ရဲ့ အပေါ်က **Parent Box** ကိုကျော်သွား တာကိုဖြင့်ရမှာ ဖြစ်ပါတယ်။ အဲတော့ ဆိုလိုချင်တာက **absolute** ကို သာမန်အတိုင်း သုံးလိုက်ရင် ကိုယ့်ရဲ့ **Parent Box** နဲ့ မသက်ဆိုက်တော့ပဲ နေရာအကွာအဝေးကို Browser Box ကနေ အကွာအဝေး စပြီးတိုင်းတာ သွားမှာဖြစ်ပါတယ် **top: 0px** နဲ့ **left: 0px** ဖြစ်တဲ့အတွက် Browser ရဲ့ အပေါ်နဲ့ဘက်ဘက်မှာ Zero Value အနေနဲ့ ကပ်သွားမှာဖြစ်ပါတယ်။

ညာဘက်က **absolute Box** အတွက် **top: 0px** နဲ့ **right: 0px** ပေးလိုက်တဲ့ အတွက် Browser ရဲ့ အပေါ်နဲ့ ညာဘက်ကို ကပ်သွားတာကို တွေ့ဖြင့်ရမှာဖြစ်ပါတယ်။ သို့ပေါ့ယယ် နောက်တစ်မျက်နှာမှာ ဒီ တန်ဖိုးတွေအတိုင်းပဲ ထားထားပြီး **Parent Box** တွေကို **relative** ပေးလိုက်မယ်ဆိုရင် ... !!



Box တန်ဖိုးတွေ ဒီအတိုင်းပဲ ထားထားပြီး Parent Box တွေကို **relative** ပေးလိုက် တယ့်အခါမှာ သက်ဆိုင်သော Parent Box တွေအတွင်းမှာပဲ တန်ဖိုးတွေ က သက်ရောက်သွားမှာဖြစ်ပါတယ်။ **position: absolute** အခန်းမှာ ဒီအချက်ကို သတိထားခေါ်ပါတယ်။

နောက် တစ်မျက်နှာမှာ ကျွန်တော်တို့ **position: fixed** ကိုလေ့လာသွားကြပါမယ်။

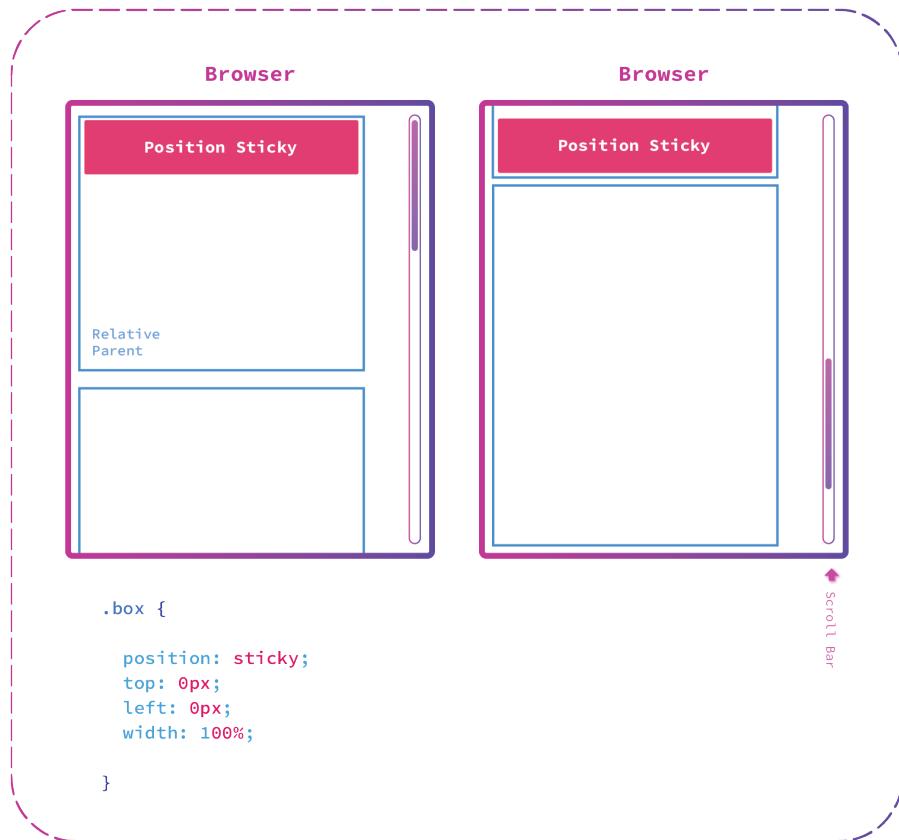


position: fixed ကိုပေးလိုက်တဲ့အခါ သက်ဆိုင်ရာ Parent အံ့ဩရောင် Box နဲ့ မဆိုင်တော့ပဲ အပြင်မှာရှိတဲ့ Browser Box နဲ့တွက်သွားမှာဖြစ်ပါတယ်။ ဆိုလိုချင်တာက Parent Box နဲ့ မသက်ဆိုက်တော့ပဲ Browser ရဲ့ အပေါ်နဲ့ဘက်ဘက်မှာ Zero Value အနေနဲ့ကပ်သွားမှာဖြစ်ပါတယ်။

position: fixed ဆိုတဲ့အတိုင်း အပေါ်မှာ Fixed ဖြစ်ပြီး ညာဘက်မှာရှိတယ် Scroll Bar ကနေ အောက်ကို စွဲချလိုက်တဲ့အခါ အနီရောင် Box လေးက Browser ရဲ့ ထိပ်ပိုင်းမှာ ကပ်လျက် လိုက်ပါလာတာကို တွေ့ရမှာပါ။

အပြင်လက်တွေ့ Websites များရဲ့ App Bar က နေရာမှာ အသုံးပြု များလေ့ရှိကြောင်း ကိုတွေ့ရမှာဖြစ်ပါတယ်။

နောက် တစ်မျက်နှာမှာ ကျွန်ုတ်တို့ fixed နဲ့ ခပ်ဆင်ဆင်တူတဲ့ position: sticky ကို လေ့လာသွားကြပါယယ်။



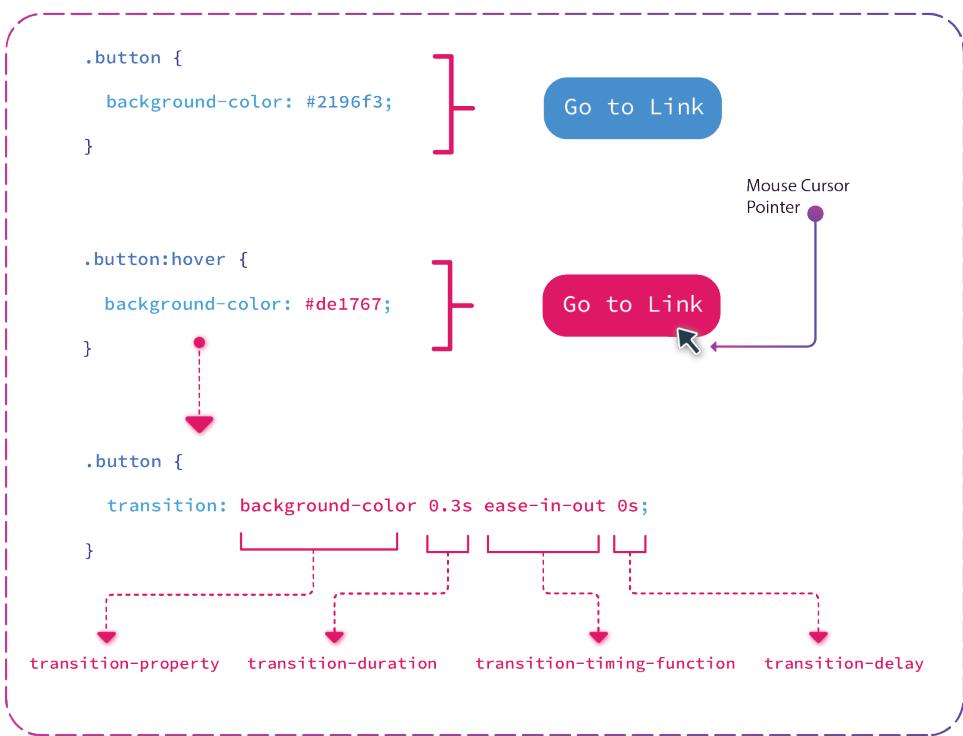
`position: sticky` ကိုပေးလိုက်တဲ့အခါ `Parent Box` နဲ့ သက်ဆိုင်မှုရှိပါတယ်။ ပြီးတော့ ညာဘက်မှာရှိတယ် Scroll Bar ကနေ အောက်ကို ဆွဲချေလိုက်တဲ့အခါ အနဲ့ရောင် Box လေးက Browser ရဲ့ ထိပ်ပိုင်းမှာ ကပ်လျက် လိုက်ပါလာတာကို တွေ့ရမှာ ဖြစ်ပါတယ်။

သို့ပေါ်မယ် သူက `Parent Box` ကနေ ရှိသလောက် Container ရဲ့ Size ပေါ်မှာပဲ ကပ်ကျွန်းနေခဲ့မှာဖြစ်ပါတယ်။ ဒါကို ဒုတိယ ပုံမှာ ကြည့်လိုက်ပါ အောက်အထိဘွားလိုက်တယ်အခါ၍ Sticky Component က အပေါ်က `Parent Box` မှာပဲကပ်လျက်ကျွန်းရှိနေခဲ့မှာဖြစ်ပါတယ်။

အခုခံ့ရင် စာဖတ်သူတို့က Position အခန်းကိုလေ့လာပြီးဖြစ်ပါတယ်။

CSS Transition များ

CSS Transition ဆိတ် ကျွန်ုပ်တော်တို့ Element ကို Action တရာ့လုပ်လိုက်တဲ့အခါန် မှာ အရောင်းပြုတယ်။ CSS Transition တွေတို့ မလုပ်ခင် pseudo class ဖြစ်တဲ့ :hover ကို မှတ်မိုက်ပြီးမယ်ထင်ပါတယ်။ :hover က Mouse Cursor (မြှား) အပေါ်တင်လိုက်တဲ့ အခါန်မှ action တစ်ခုရလုပ်ချင်တဲ့အခါအသုံးပြုတာဖြစ်ပါတယ်။



ပထမ (၂) ခုက ကျွန်ုပ်တော်တို့ သိပြီးသား :hover Cursor တင်လိုက်တဲ့အခါ အပြောဂျင် ခလုတ်လေးကနေ အနိုရောင် ပြောင်းသွားတာကို ရေးသားထားခြင်းဖြစ်ပါတယ်။ ဒေါက်ဆုံး တစ်ခုက transition အသုံးပြုပုံကိုရေးသားထားတာဖြစ်ပါတယ်။

ပထမဆုံး တန်ဖိုးက transition-property : background-color အဖြစ်ထည့်သားတာ ကိုမြင်ရမှာပါ။ အဓိပ္ပာယ်က background-color အတွက် transition ပြုလုပ်မယ်လို့ ရေးသားထားခြင်းဖြစ်ပါတယ်။

ဒုတိယ တန်ဖိုးက transition-duration : 0.3s ကြောချိန်က 0.3 စဲ့နဲ့ အတွင်း Animation ပြုလုပ်မယ်လို့ သတ်မှတ်ထားတာဖြစ်ပါတယ်။

တတိယ တန်ဖိုးက transition-timing-function ကိုဆိုလိုတာဖြစ်ပါတယ်။ အထဲက တန်ဖိုးတွေမှာ linear, ease-in-out, ease-in, ease-out တွေက အသုံး များပါတယ်။ Animation ပြုလုပ်တဲ့အချိန်မှာ linear ဆိုတာက တသတ်မှတ်တည်း အရိုန်နဲ့သွားမယ်လို့ ဆိုလိုတာပါတယ်။ ease-in-out ဆိုတာက အဝင်နဲ့အထွက်မှာ အရိုန်နေးမြင် Smooth ဖြစ်အောင်လုပ်ပြီး အလယ်ပိုင်းမှာတော့ မြန်သွားပါမယ်။ ease-in ဆိုတာက အစအဝင်မှာ အရိုန်နေးပါမယ်။ ease-out ဆိုတာက အထွက်မှာ အရိုန်နေးပါမယ်။ စာဖတ်သူတို့က timing-function တွေကို ဒီလောက်နားလည်ရင် ရပါပြီ။ အသေးစိတ်ကတော့ ထပ်ပြီးလေ့လာသွားရမှာပါ။

စတုထွေတန်ဖိုးကတော့ transition-delay အချိန်ပဲဖြစ်ပါတယ် စာဖတ်သူတို့က Action တစုစလုပ်လိုက်တဲ့အချိန်မှာ ချက်ချက် Animation စမလုပ်ပဲ ခကေတ္တင့်စေချင်တယ် Delay ဖြစ်စေချင်တဲ့အခါ အသုံးပြုလိုရပါတယ်။

အခုပြထားတဲ့ Example အရ ခလုတ်လေးပေါ့မှာ Cursor တင်လိုက်တဲ့အခါ ချက်ချင်း အနီးရောင်ကို ပြောင်းသွားမှာမဟုတ်တော့ဘဲ 0.3s အကြား ပေးဖော်ချင်း အနီးရောင် ပြောင်းသွားမှာကိုမြင်ရမှာပါ။ နောက်ထပ် ဥပမာ တစ်ခုကို အောက်ကပုံတွင် မြင်သာ အောင် လေ့လာကြည့်ရအောင် ...။

```
.box {
    width: 80px;
    height: 80px;
    transition: all 0.3s ease-in-out;
}
.box:hover {
    width: 120px;
    height: 120px;
}
```

အကျယ် Width နဲ့ Height 80px အရွယ်အစားရှိတဲ့ Box တဲ့ 120px အနေနဲ့ ပြောင်းလည်းလိုက်တာဖြစ်ပါတယ်။ ချက်ချင်းရှုပ်တရက်စပြာင်းသွားတာမဟုတ်ပဲ 0.3 second အတွင်းမှာ Smooth ဖြစ်စာပြားလဲသွားမှာပါ။

တမျက်နှာ - 203

CSS Background Image အကြောင်း

စာဖတ်သူတိန္ဒြေ အတူတူ background-color ကို ဥပမာတော်တော်များများမှာ အသံးပြု ခဲ့ကြပါတယ်။ အခြေအနေမှာတော့ အခြား background-image နဲ့ပါတ်သတ်တဲ့ အကြောင်းအရာများကို လေ့လာသွားကြပါမယ်။

Original Image - leave.jpg ကို အသံးပြု။
ပြီး Background Image အနေနဲ့
သတ်မှတ်မှုကြပါတယ်။

```
.container {
    background-image: url("leaves.jpg");
    background-size: auto;
    background-repeat: repeat;
    background-position: 50% 50%;
}
```

background-image ဆိုတာက Image ရဲ့နေရာကို ထည့်ပေးရမှုဖြစ်ပါတယ်။

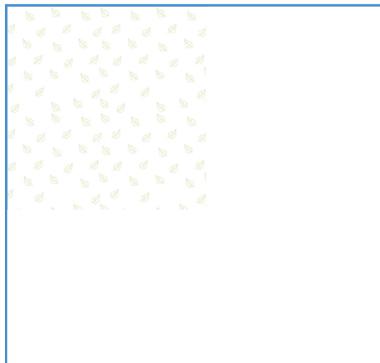
background-size ဆိုတာက ထည့်ထားတဲ့ image ရဲ့ Size ကိုပြောင်းလဲစိုးအတွက်ဖြစ်ပါတယ်။

background-repeat ဆိုတာကတော့ ပုံထပ်တလဲလဲပြန်သုံးချင်တဲ့အခါ သုံးလို့ရပါတယ်။

background-position ဆိုတာကတော့ image ရဲ့ အနေအထား နေရာကို သတ်မှတ်ချင်တဲ့အခါ အသံးပြုပါတယ်။

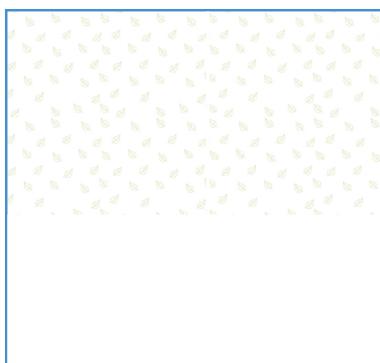
```
.container {  
    background-image: url("leaves.jpg");  
    background-repeat: no-repeat; ●
```

background-repeat ကို no-repeat လိုပေးလိုက်
တဲ့အခါ ငှက ရှိသလောက်ပဲ Box ထဲမှာပေါ်တော့
မှာဖြစ်ပါတယ်။



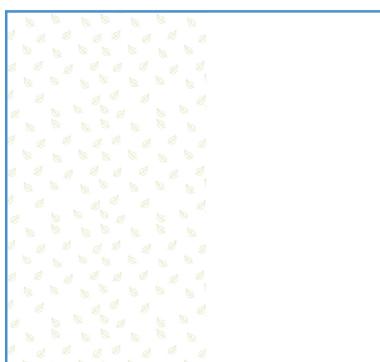
```
.container {  
    background-image: url("leaves.jpg");  
    background-repeat: repeat-x; ●
```

background-repeat ကို repeat-x လိုပေးလိုက်တဲ့
အခါ ငှက x ဝင်ရှိးဖြစ်တဲ့ အလျားလိုက် ထပ်ခါတ
လဲလဲ ပေါ်ဘွားမှာဖြစ်ပါတယ်။



```
.container {  
    background-image: url("leaves.jpg");  
    background-repeat: repeat-y; ●
```

background-repeat ကို repeat-y လိုပေးလိုက်တဲ့
အခါ ငှက y ဝင်ရှိးဖြစ်တဲ့ ဒေါင်လိုက် ထပ်ခါတလဲ
လဲ ပေါ်ဘွားမှာဖြစ်ပါတယ်။

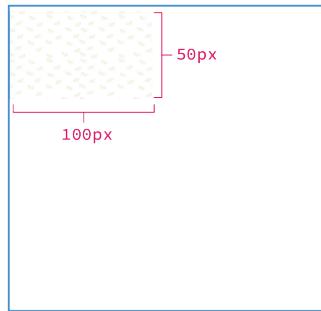
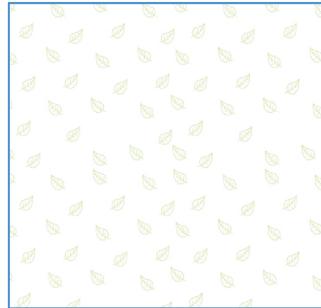


```
.container {
    background-image: url("leaves.jpg");
    background-repeat: no-repeat;
    background-size: cover; ●
}
```

no-repeat ပေးထားသော်လည်း background-size ကို cover ထားလိုက်ပဲအခါမှာ ပုံတစ်ခုလုံးကို ဖြန့်ကျပြီးတော့ အပြည့် ထားထားပေးမှာဖြစ်ပါတယ်။

```
.container {
    background-image: url("leaves.jpg");
    background-repeat: no-repeat;
    background-size: 100px 50px; ●
}
```

background-size ကို x တန်ဖိုးနဲ့ y တန်ဖိုးအနေနဲ့
လည်း သတ်မှတ်ပေးလို့ရပါတယ်။

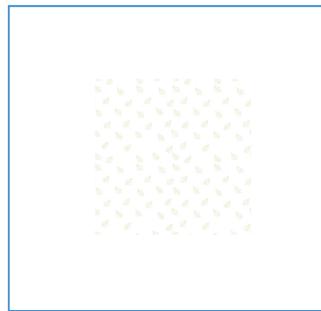


```
.container {
    background-image: url("leaves.jpg");
    background-repeat: no-repeat;
    background-position: center center; ●
}
```

background-position: ကာစော့ image ရဲ့ နေရာကို
x တန်ဖိုးနဲ့ y တန်ဖိုးအနေနဲ့ သတ်မှတ်တာဖြစ်ပါတယ်
။ x - center ရယ် y - center ရယ်မလို့ ပုံကအလေယ်
တည်ရှုပါတယ်။

```
.container {
    background-image: url("leaves.jpg");
    background-repeat: no-repeat;
    background-position: top right; ●
}
```

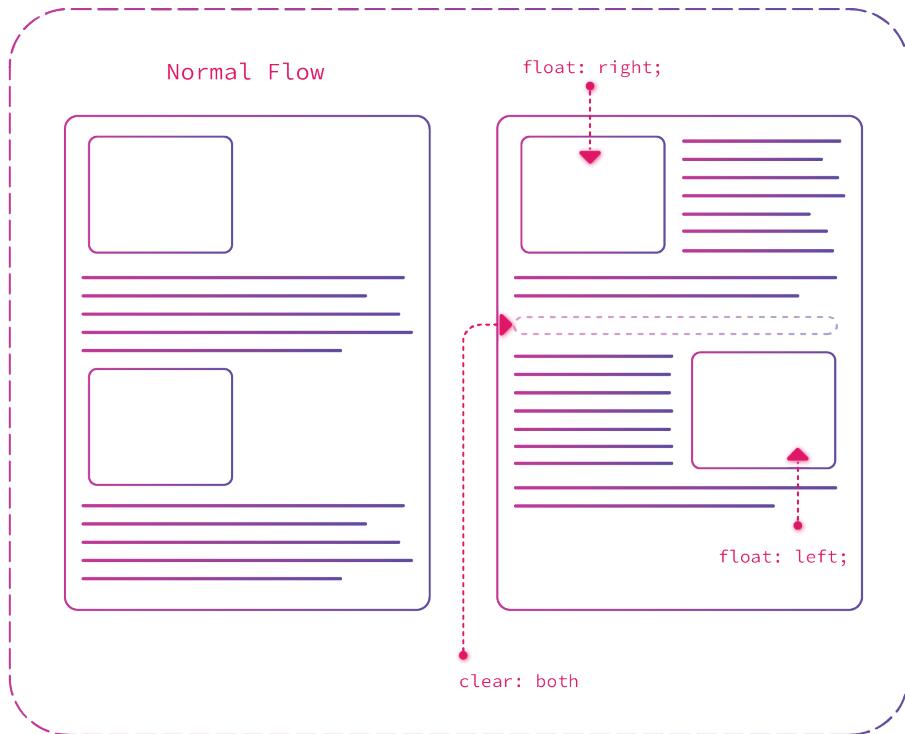
top ရယ် right ရယ်ကြောင့် ပုံက အပေါ်ညာဘက်မှာ
ပေါ်နေမှာပါ။ percentage (%) နဲ့လည်းအသုံးပြုနိုင်
ပါတယ်။



CSS ဖွင့် Layout များကို လွှဲလာခြင်း

CSS Float အကြောင်း

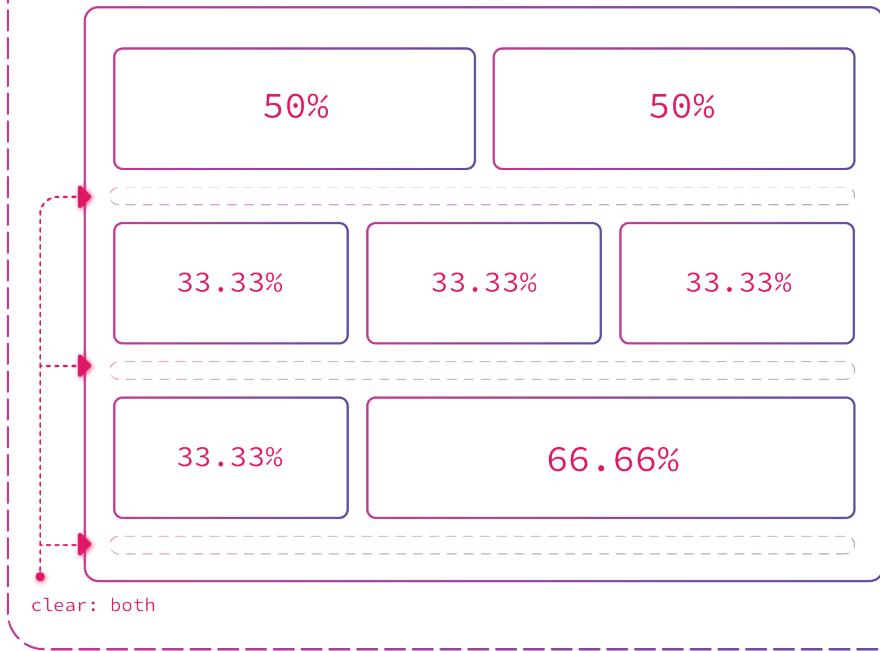
Modern ခေတ် CSS Layout အပိုင်းမှာ Float က အသံအရမ်းနည်းသွားပါပြီ သို့ပေါ့
မယ့်လည်း ကျွန်တော်တို့ သိသင့်ပါတယ်။ Project အဟောင်းတွေ ပြန်ပြင်ရတဲ့ အခါ ဒီ
လိုပုံစံတွေကို ပြန်လည်ပြင်ဆင်ရတာမျိုးတွေလည်းရှိတတ်ပါတယ်။ အောက်ကပုံလေး
ကို တစ်ချက်ကြည့်ပေးပါ။



Normal Flow အရ Image တွေက Inline Element ဖြစ်ပါတယ် နောက် အောက်က
စာကြောင်းတွေက <p> Element လို့သဘောထားလိုက်ပါ သူတို့က နောက်တစ်လိုင်း
ဆင်းသွားပါလိမ့်မယ်။ ညာဘက်က ပုံကို ကြည့်ကြည့်လိုက်ပါ Image ကို float : right
ဆိုပြီး ကြော်ပေးလိုက်တဲ့ အခါမှာ Page ရဲ့ Flow က ပျက်သွားပြီး စာသားတွေက
ညာဘက်ကို ရောက်သွားပါလိမ့်မယ်။ ဆိုလိုချင်တာက float: right သုတိက်တဲ့ အခါ
နောက်ထပ် Element တွေအားလုံးက ညာဘက်ကိုပဲ Floating ဖြစ်ပြီး ရောက်သွားပါ
လိမ့်မယ်။ အဲ နောက်ထပ် Element တွေကို Floating Effect ကို ပြန်ဖျက်ပေးဖို့အတွက်
clear ကိုသုံးလိုရပါတယ်။ clear: right ဆိုရင် float: right ကိုရှင်းလင်းပေးတာဖြစ်ပြီး
နှစ်ဖက်လုံးကို ဖျက်ချင်ရင် clear: both ကိုအသုံးပြုနိုင်ပါတယ်

ပုံထဲမှာတော့ both ကိုပဲသုံးပြီး ရှင်းလိုက်ပါတယ်။ နောက်ထက် အောက်က Image Element ကိုတော့ float:left ဆိုပြီး ကြော်လိုက်တဲ့အပါ ဘသားတွေက ဘယ်ဘက်မှာ Floating Effect ကိုသက်ရောက်သွားတဲ့အတွက် ဘယ်ဘက် တစ်ချက်ဆီမှာပဲ ရှိတော့ မှာပါ။ ဒီလောက်ဆို Float အသုံးပြုပဲကို စာဖတ်သူတို့က အနည်းအကျဉ်းတော့ သိသွားပြီလို့ယူဆလိုက်ပါမယ်။ အရင်အချိန်တွေတိုန်းက float ကိုသုံးပြီး layout ပြုလုပ်ပဲကို အောက်ကပုံလေးက တစ်ဆင့်လေ့လာကြည့်ရအောင် ...။

Float Layout



အထဲက လေးထောင့် Box တွေအကျိန်လုံးကို float:left လို့ကြော်လိုက်ပြီး width ကို Percent လေးတွေနဲ့ လိုက်ကစားလိုက်မယ်ဆိုရင် ပုံထဲကအတိုင်း ကျွန်ုပ်တော်တို့က Layout ကို ချထားလို့ရပါတယ်။ နောက် တစ်ကန့် Box တွေကို layout မချခင် ကြားထဲမှာ clear:both ဆိုပြီး Floating Effect ကိုပြန်ဖျက်ကာ နောက်တစ်တန်းအတွက် ပြန်ပြီး Layout ပြန်ချလို့ရပါတယ်။ ဒီလိုနဲ့ Webpage တစ်ခုလုံးကို လိုချင်သလို ပုံစံခွက်လေးထဲဝင်အောင် စာဖတ်သူတို့က အသုံးပြုနိုင်ပါတယ်။

CSS Float အကြောင်းကို စာဖတ်သူတို့ လေ့လာခဲ့ပြီးဖြစ်ပါတယ်။

အဲတော့ Coding ပိုင်းအရစမ်းလိုက်အောင် ဖိုလ်ဒါအသစ် “27_css_layout_float” ဆိုပြီး တည်ဆောက်လိုက်ပါမယ်။ “27_css_layout_float” ဆိုတဲ့ ဖိုလ်ဒါ အတွင်းထဲမှာ “index.html” နဲ့ “style.css” ဆိုပြီး HTML နဲ့ CSS ဖိုင် (J) ခုတည်ဆောက်လိုက်ပါမယ်။

စာဖတ်သူတို့က index.html တွင် ဒီအောက်က Code ကိုကူးယူပြီးစမ်းကြည့်လို့ရပါမယ်။

<https://gist.github.com/lwinmoepaing/02240b34000d74fec011d1b98b51b9d4>

style.css တွင် ဒီအောက်က Code ကိုကူးယူပြီးစမ်းကြည့်ပါ။

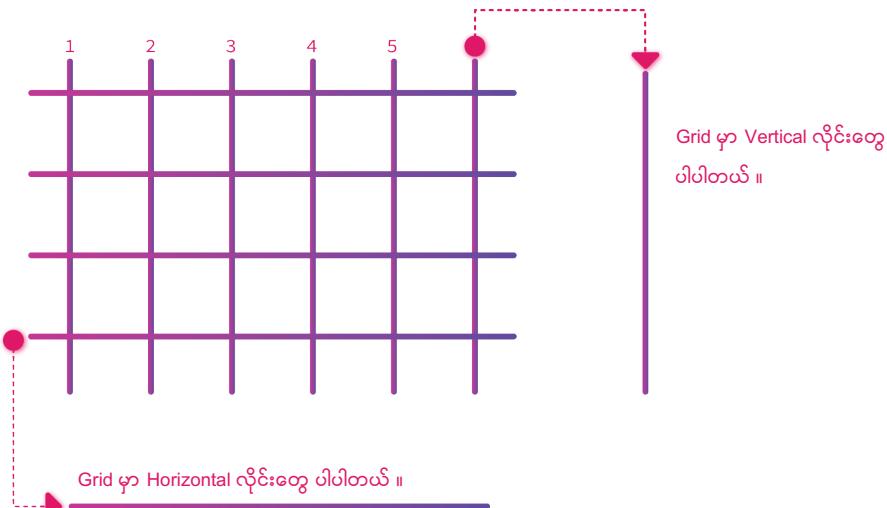
<https://gist.github.com/lwinmoepaing/de6b511505cfdb3d14923af5bf54f9fc>

Codepen မှာလည်းတိုက်ရှိက် စမ်းကြည့်နိုင်ပါတယ်။

<https://codepen.io/lwinmoepaing/pen/jOXEzBJ>

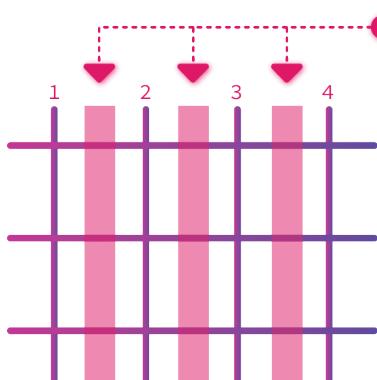
CSS Grids အကြောင်း

ဒီအပိုင်းမှာတော့ ကျွန်ုတ်တို့ Layout တွေကို ပုံစံချေတဲ့အပါ Modern ခေတ်နောက်ပိုင်း အသုံးအချခံး Grid CSS ဖြစ်ပါတယ်။ Grid Structure မှာဘာတွေပါလဲဆိုတာကို အောက်ကပုံလေးကို အရင်လေ့လာကြည့်ရအောင်။

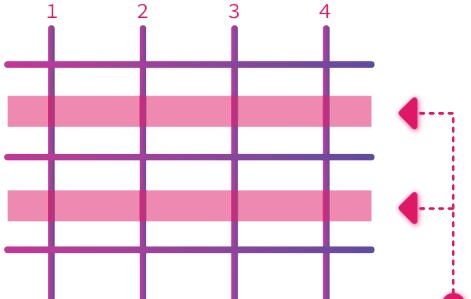


Horizontal လိုင်းကော့ Vertical လိုင်းကော့ပါတဲ့အတွက် Two Dimensional အတွက် Directions လိုသလို သူက Layout ကိတ်နှုန်းချုပ်နိုင်ပါတယ်။

ကျွန်တော်တို့ Table အခန်းမှာ လေ့လာထားခဲ့ကြသလိုပဲ Grid မှာ Row နဲ့ Column တွေကို ဘယ်လိုနေရာယူလဲကို လေ့လာကြည့်ရအောင် ... !!



Grid မှာ Vertical လိုင်းကြားထဲတွင်ရှိတဲ့ နေရာတွေကို Column လိုပေါ်တယ် !!



Grid မှာ Horizontal လိုင်းကြားထဲတွင်ရှိတဲ့ နေရာတွေကို Row လိုပေါ်တယ် !!

- Grid Column ကိုစလေ့လာခြင်း

စာဖတ်သူတို့က [index.html](#) တွင် ဒီအောက်က Code ကိုကူးယူပြီးစမ်းကြည့်လိုပါမယ် !!

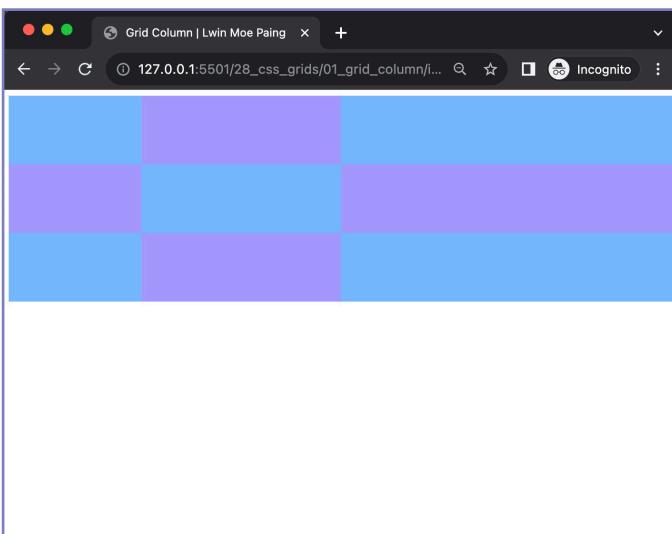
<https://gist.github.com/lwinmoepaing/80d509f2c671a47c83aac8d8105bd6ca>

[style.css](#) တွင် ဒီအောက်က Code ကိုကူးယူပြီးစမ်းကြည့်ပါ !!

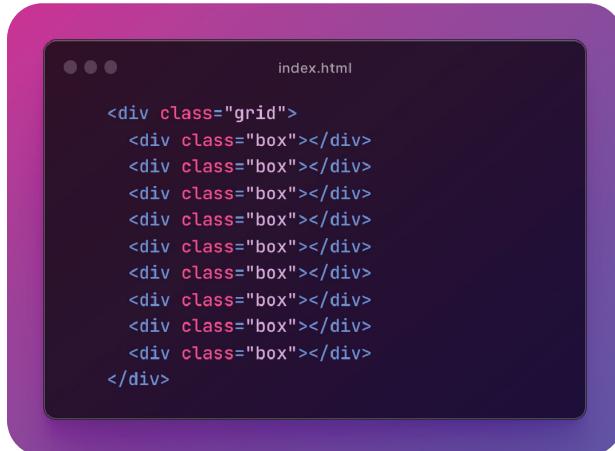
<https://gist.github.com/lwinmoepaing/1621f9265368d813f580109a6c447c48>

Codepen မှာလည်းတိုက်ရိုက် စမ်းကြည့်နိုင်ပါတယ် !!

<https://codepen.io/lwinmoepaing/pen/OJrXWEX>

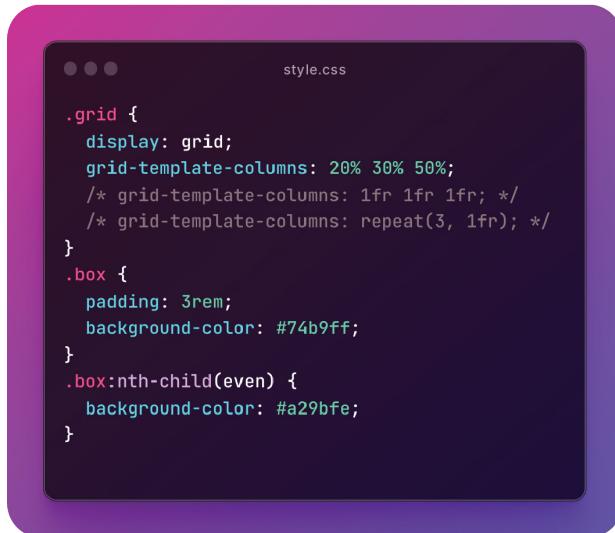


Browser မှာ စာဖတ်သူကိုက Run လိုက်တဲ့အခါမှာ အပြောကွက်လေးတွေနဲ့ ခရမ်းကွက် (၉) ကွက်ကို မြင်ရမှာပါ။ ဒေါက်က Code အတိုင်း container ကို "grid" class နဲ့ ရေးသားပြီးတော့ အတွင်းပိုင်းမှာ "box" class ကိုခုကိုရေးသားထားတာဖြစ်ပါတယ်။



```
index.html
<div class="grid">
  <div class="box"></div>
  <div class="box"></div>
</div>
```

CSS ရေးသားပုံကို လေ့လာကြည့်ရအောင် ...။

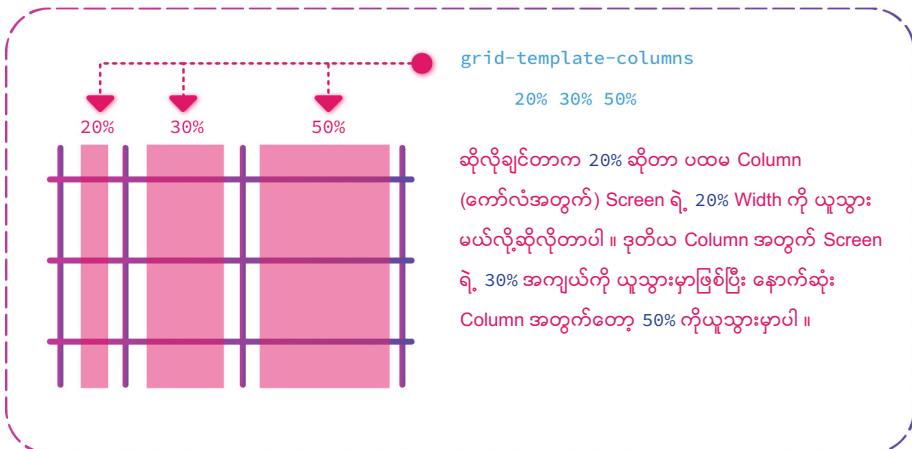


```
style.css
.grid {
  display: grid;
  grid-template-columns: 20% 30% 50%;
  /* grid-template-columns: 1fr 1fr 1fr; */
  /* grid-template-columns: repeat(3, 1fr); */
}
.box {
  padding: 3rem;
  background-color: #74b9ff;
}
.box:nth-child(even) {
  background-color: #a29bfe;
}
```

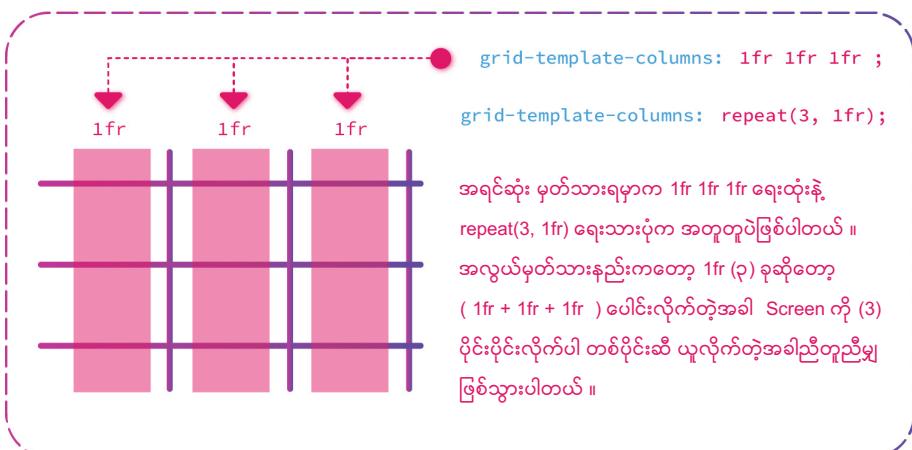
`display: grid;` ဆိုပြီးရေးသားထားတာကို သတိပြုပေးပါ။ ကျွန်ုတ်တို့ အရင်တုန်းက လေ့လာထားတဲ့ inline ရယ် block level element နှစ်မျိုးရှိပါတယ်။ အခုခြားမှာ ကတော့ Grid layout အသုံးပြုဖို့အတွက် "`display: grid`" ဆိုပြီး container Element က သတ်မှတ်ထားတာဖြစ်ပါတယ်။ `.box` တွေက အပြောရောင်ဖြစ်ပြီး (စုံ-ဝက်နှင့်) `.box` တွေ ကတော့ ခရမ်းရောင်အဖြစ်ရေးသားထားတာဖြစ်ပါတယ်။

`grid-template-columns` ဆိတာက Grid Columns တွေအတွက် ဘယ်လိုပုံစံ Layout ကို စီမံမယ်ဆိုပြီးသတ်မှတ်တဲ့အခါအသုံးပြုပါတယ်။

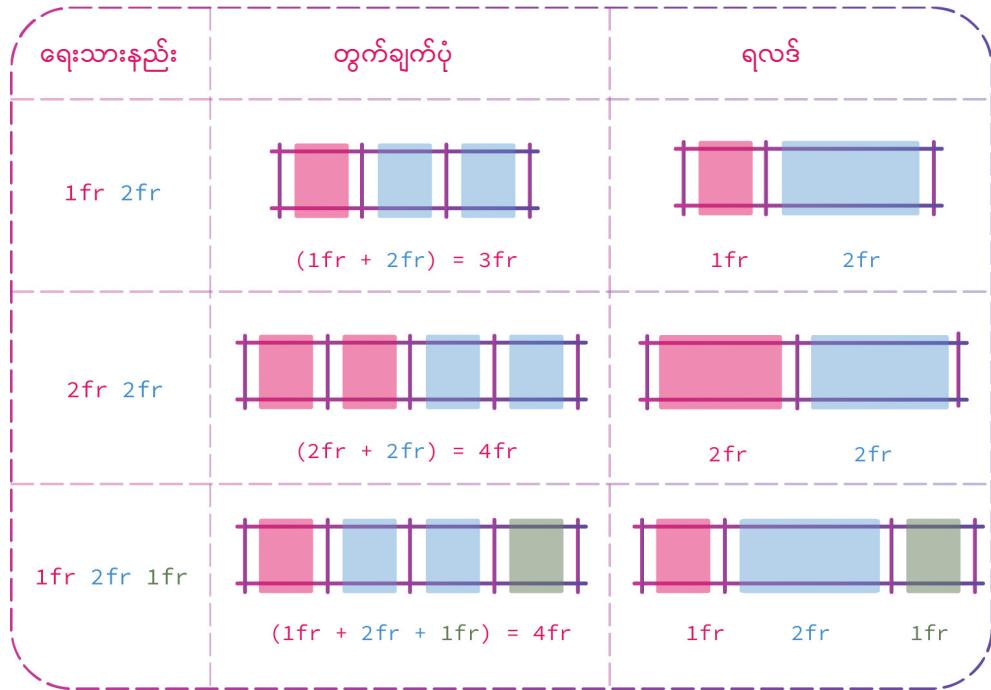
ပထမဆုံး `grid-template-columns: 20% 30% 50%`; ရေးသားသတ်မှတ်ပုံကိုလေ့လာ ကြည့်ရအောင် ...။



စာဖတ်သူတို့က `grid-template-columns: 1fr 1fr 1fr`; ငြောင်းလည်း ရေးသားကြည့်ပါ မိအခါ Browser မှာ Column (ဂုဏ်) ခုစုံတဲ့က ညီတူညီမှု အနေနဲ့ ပေါ်လာမှာကို မြင်ရမှာ ဖြစ်ပါတယ်။ `fr` ကို Fractional Unit လို့လည်းခေါ်ပါတယ်။



အခါ Browser မှာ Run လိုင်တဲ့အခါ အဲဒ္ဓအတိုင်း ညီတူညီမှုယူသွားမှာကို တွေ့ရမှာဖြစ်ပါတယ်။ `fr` အကြောင်းကိုပိုနားလည်အောင် နောက်တစ်မျက်နှာမှာ ပုံနဲ့တကွ လေ့လာ ကြည့်ပါမယ်။



ရေးသားနည်းထက် fr တစ်ခုချင်းဆိုကိုပေါင်းပြီး (တွက်ချက်ပုံ) ထက်အတိုင်း အပိုင်းပိုင်း ချလိုက်ပါ 3fr ဆို 3 ပိုင်းပိုင်းချလိုက်ပါ။ 4fr ရလျှင် 4 ပိုင်းပိုင်းချလိုက်ပါ။ ပြီးမှ သူရှိ သလောက်အပိုင်းအတိုင်း နေရာယူလိုက်တဲ့အခါ ညာဘက်က (ရလဒ်) အတိုင်း Browser ပေါ်တွင်ရရှိလာမှာဖြစ်ပါတယ်။

- Grid Rows ကို လေ့လာခြင်း

ကျွန်တော်တို့ Columns အကြောင်းကိုလေ့လာပြီးသွားတဲ့အခါ Rows တွေအကြောင်း ကို ဆက်လက်လေ့လာကြပါမယ်။

စာဖတ်သူတို့က [index.html](#) တွင် ဒီအောက်က Code ကိုကူးယူပြီးစမ်းကြည့်လို့ရပါမယ်။

<https://gist.github.com/lwinmoepaing/60bd9551cd728f641d4609dc8d399181>

[style.css](#) တွင် ဒီအောက်က Code ကိုကူးယူပြီးစမ်းကြည့်ပါ။

<https://gist.github.com/lwinmoepaing/51eb7f6b13777e0d970ee2a28160bcd3>

Codepen မှာလည်းတိုက်ရိုက် စမ်းကြည့်နိုင်ပါတယ်။

<https://codepen.io/lwinmoepaing/pen/QWzQoeK>

CSS ရေးသားပုံကို လေ့လာကြည့်ရအောင် ... ။

```

style.css

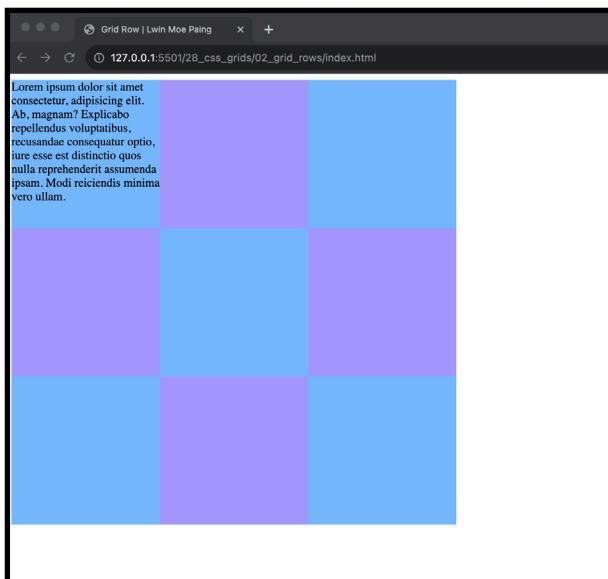
.grid {
    display: grid;
    grid-template-columns: repeat(3, 200px);
    grid-auto-rows: 200px;
    /* grid-auto-rows: minmax(200px, auto); */
    /* grid-template-rows: 100px 200px 300px; */
    /* grid-template-rows: repeat(3, minmax(50px, auto)); */
}

.box {
    background-color: #74b9ff;
}

.box:nth-child(even) {
    background-color: #a29bfe;
}

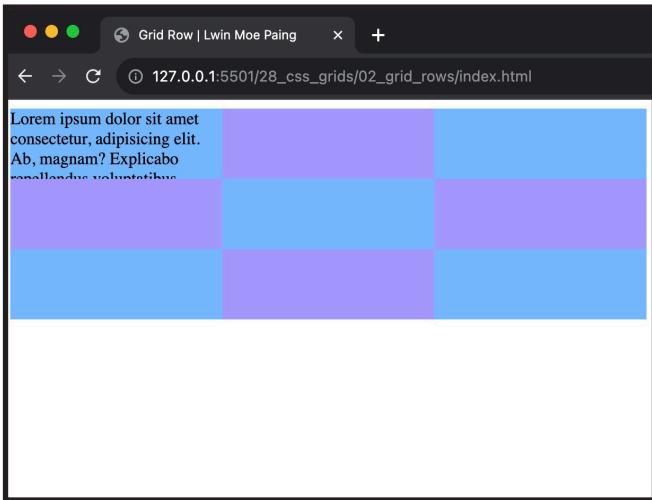
```

`display: grid;` ဆိုပြီးရေးသားထားတာကို သတိပြုပေးပါ။ `.box` တွေက အဲပြာရောင်ဖြစ်ပြီး (ခုံ-ကဗျာန်း) `.box` တွေကတော့ ခရိုင်းရောင်အဖြစ်ရေးသားထားတာဖြစ်ပါတယ်။ `grid-template-columns: repeat(3, 200px);` ရေးသားထားခြင်းက အရင် Columns အခန်းမှ လေ့လာပြီးသလိုမျိုး Col (ဂုဏ်ပိုင်) ခုအတွက် `200px` အညီအမှုရှိနေမှာဖြစ်ပါတယ်။



Rows အတွက်ရေးသားထားပုံမှာ `grid-auto-rows: 200px` ဖြင့် ရေးသားထားခြင်းကို သတိပြုကြည့်ပါ။ သူက Auto အနေနဲ့ Row တစ်ခုချင်းဆီတိုင်းကို 200px သက်ရောက်သွားမှာဖြစ်ပါတယ်။ အဲဒါကြောင့် အောက်က Box တစ်ခုချင်းဆီတိုင်းက 200px height (အမြင့်) ကို ရရှိသွားမှာပါ။

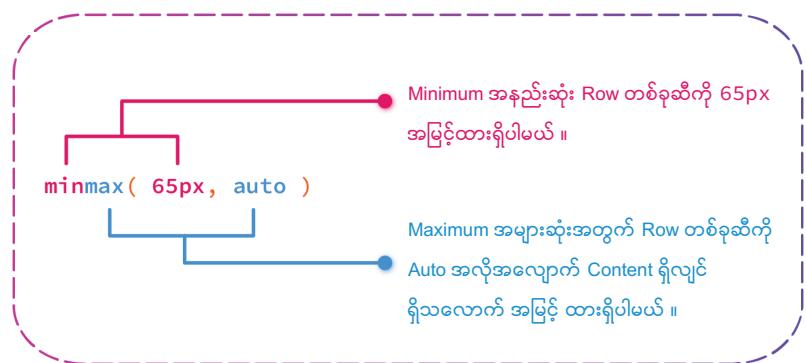
`grid-auto-rows: 65px;` ဖြင့် ဖြောင်းလဲ ရေးသားကြည့်လိုက်တဲ့အပါမှာ အောက်က ပုံ အတိုင်း ဖြစ်သွားမှာပါ။



ရေးသားကျွန်ုတ်တို့ ပထမ Box အတွင်းမှာရေးသားထားသော စာမျက်းက အထဲသို့ နစ်မြုပ်သွားတာကို မြင်ရမှာဖြစ်ပါတယ်။ တကယ်တော့ အထဲက Content ဘသားများ အတွက် 65px ထက်ပို့တဲ့ အမြင့် (height) ကိုလိုအပ်ပါတယ်။ သို့ပေါ့မယ်

`grid-auto-rows: 65px;` ကိုပေးလိုက်တဲ့အပါ Box တွေအားလုံးက 65px ကိုပဲရရှိသွားတဲ့အတွက်ဖြစ်ပါတယ်။ ဒီလိုဘသားများ ပျက်သွားတာမလိုချင်တဲ့အတွက် ကျွန်ုတ်တို့က ရေးသားနည်းနောက်တစ်မျိုး အသုံးပြုနိုင်ပါတယ်။

`grid-auto-rows: minmax(65px, auto);` အောက်ကပုံကိုကြည့်ပေးပါ။

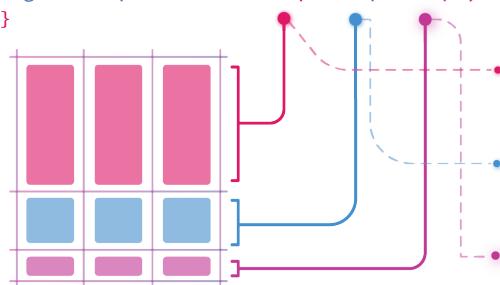




grid-auto-rows: `minmax(65px, auto)`; ရေးသားပုံကြောင့် အပေါ်က စာသား အတွင်းရှိသလောက် Auto ချိန်ညိုပေးသွားပြီး၊ ကျွန်ုရီနောက် Rows တွေကတော့ 65px ကို ရရှိနေမှာပဲဖြစ်ပါတယ်။

ဒီတစ်ခါ grid-auto-rows ကိုအသုံးမပြုတော့ပဲ grid-template-rows ကို အသုံးပြုသွားကြပါမယ်။ grid-template-rows ကလည်း Grids Rows တွေအတွက် ဘယ်လိုပုံစံ Layout ကိစ္စမယ်ဆိုပြီးသတ်မှတ်တဲ့အခါအသုံးပြုပါတယ်။ အောက်ကပုံကို တွေ့ကြတယ်။

```
.grid {
  display: grid;
  grid-template-columns: repeat(3, 200px);
  grid-template-rows: 300px 200px 100px;
}
```

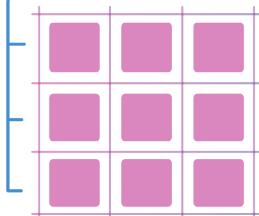


ပထမ Row အတွက် 300px
အမြင့်ထားရှိပါမယ်။

ဒုတိယ Row အတွက် 200px
အမြင့်ထားရှိပါမယ်။

တတိယ Row အတွက် 100px
အမြင့်ထားရှိပါမယ်။

```
.grid {
  display: grid;
  grid-template-columns: repeat(3, 200px);
  grid-template-rows: repeat(3, minmax(50px, auto));
}
```



Row 3 ခုအတွက် repeat ကိုအသုံးပြု
ထားပြီး။ Row တစ်ခုချင်းဆီအတွက်
Minimum အနည်းဆုံး 50px အမြင့်
ထားရှုပါမယ်။ Maximum အများ
ဆုံးအတွက် Auto အလိုအလောက် အမြင့်
Content ရှိလျှင် ရှိသလောက် အမြင့်
ထားရှုပါမယ်။

grid-template-rows ရေးသားပုံကို လေ့လာခဲ့ပြီးဖြစ်ပါတယ်။ ဒါဆိုရင် Grid တစ်ခုနဲ့
တစ်ခု ကြားအကွာအဝေးကို သတ်မှတ်ဖို့အတွက် grid gap အကြောင်းကို လေ့လာသွား
ကြပါမယ်။

- Grid Gaps ကို လေ့လာခြင်း

ကျွန်တော်တို့ Grid Rows တွေအကြောင်းကိုလေ့လာပြီးသွားတဲ့အခါ Gaps တွေ
အကြောင်းကို ဆက်လက်လေ့လာကြပါမယ်။

စာဖတ်သူတို့က [index.html](#) တွင် ဒီအောက်က Code ကိုကူးယူပြီးစမ်းကြည့်လို့ရပါမယ်။
<https://gist.github.com/lwinmoepaing/807f9e8fb083796e87a7b95dfc24a4a5>

[style.css](#) တွင် ဒီအောက်က Code ကိုကူးယူပြီးစမ်းကြည့်ပါ။

<https://gist.github.com/lwinmoepaing/274f2084a77572f264096c678ed5652e>

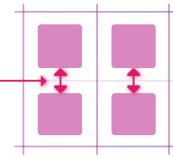
Codepen မှာလည်းတိုက်ရှိက် စမ်းကြည့်နိုင်ပါတယ်။

<https://codepen.io/lwinmoepaing/pen/YzdLgpQ>

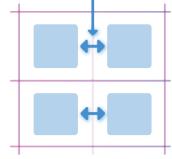
grid-row-gap: 10px; နဲ့ grid-column-gap: 10px; ရေးသားပုံကို စာဖတ်သွို့က ထွေရှိရမှာဖြစ်ပါတယ်။ ပေးထားသော Code များကို Browser ၏ Run လိုက်တဲ့အခါ မှာလည်း အကွက်လေးတွေက တစ်ခုနဲ့ တစ်ခု ကပ်မနေတော့ဘဲ ကြားထဲမှာ Spacing ကွားမျှရှိနေတာကို ထွေရမှာပဲဖြစ်ပါတယ်။ အောက်က ပုံ(၂) ပုံကို ကြည့်ပြီး လေးလာကြည့်ရအောင် ...။

```
.grid {
  display: grid;
  grid-template-columns: repeat(3, 200px);
  grid-template-rows: repeat(3, minmax(50px, auto));
}
```

grid-row-gap: 10px;
grid-column-gap: 10px;



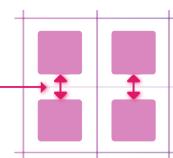
grid-row-gap ကို row တွေကြားထဲက
အကွာအဝေးကိုသတ်မှတ်ပါတယ်။



grid-column-gap ကို column တွေကြားထဲက
အကွာအဝေးကိုသတ်မှတ်ပါတယ်။

```
.grid {
  display: grid;
  grid-template-columns: repeat(3, 200px);
  grid-template-rows: repeat(3, minmax(50px, auto));
}
```

grid-gap: 50px 10px;



grid-gap ရေးသားပုံနဲ့လည်း အတိုကောက် row ရော့ column အတွက်ပါ ရေးသားနိုင်ပါတယ်။
ထူးဆုံး 50px က row အတွက်ဖြစ်ပြီး ခုတိယ 10px က col အတွက်ဖြစ်ပါတယ်။

- Grid Column Position နေရာခြင်း

စာဖတ်သူတို့က [index.html](#) တွင် ဒီအောက်က Code ကိုကူးယူပြီးစမ်းကြည့်လိုပါမယ်။

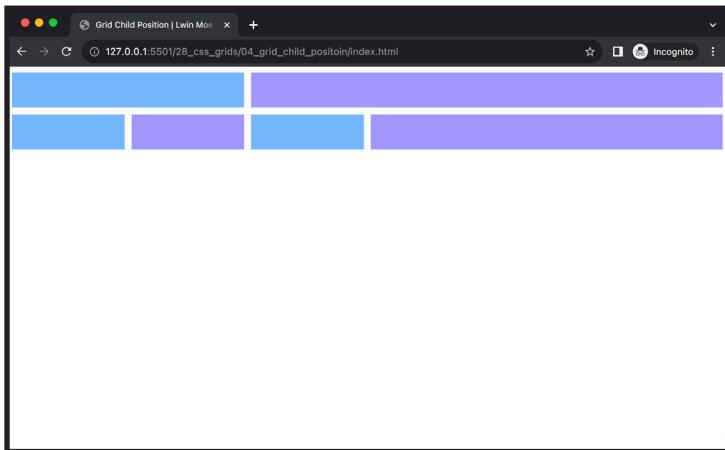
<https://gist.github.com/lwinmoepaing/6f672213258d2e90b3c81fd163ccd367>

[style.css](#) တွင် ဒီအောက်က Code ကိုကူးယူပြီးစမ်းကြည့်ပါ။

<https://gist.github.com/lwinmoepaing/0e1b6576f4c9b1c3db2e0f7e986d1be6>

Codepen မှာလည်းတိုက်ရိုက် စမ်းကြည့်နိုင်ပါတယ်။

<https://codepen.io/lwinmoepaing/pen/XWoYWGj>



Browser ပေါ်မှာ Run လိုက်တဲ့အခါ အခုလို တွေ့ရှိရမှာဖြစ်ပါတယ်။ CSS ရေးသားပုံမှာ အောက်ကပုံအတိုင်း ရေးသားထားပါတယ်။

```
styles.css

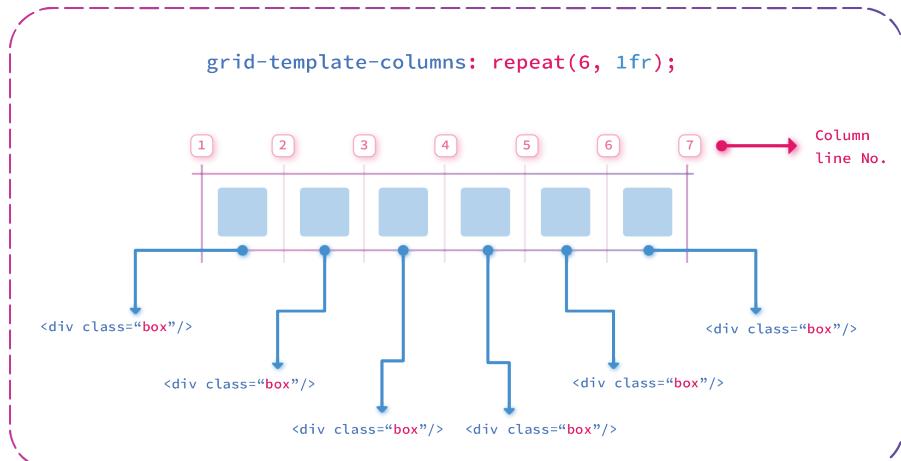
.grid {
    display: grid;
    grid-template-columns: repeat(6, 1fr);
    grid-template-rows: repeat(7, minmax(50px, auto));
    grid-gap: 10px;
}

.take-2 {
    grid-column: 1 / 3;
}

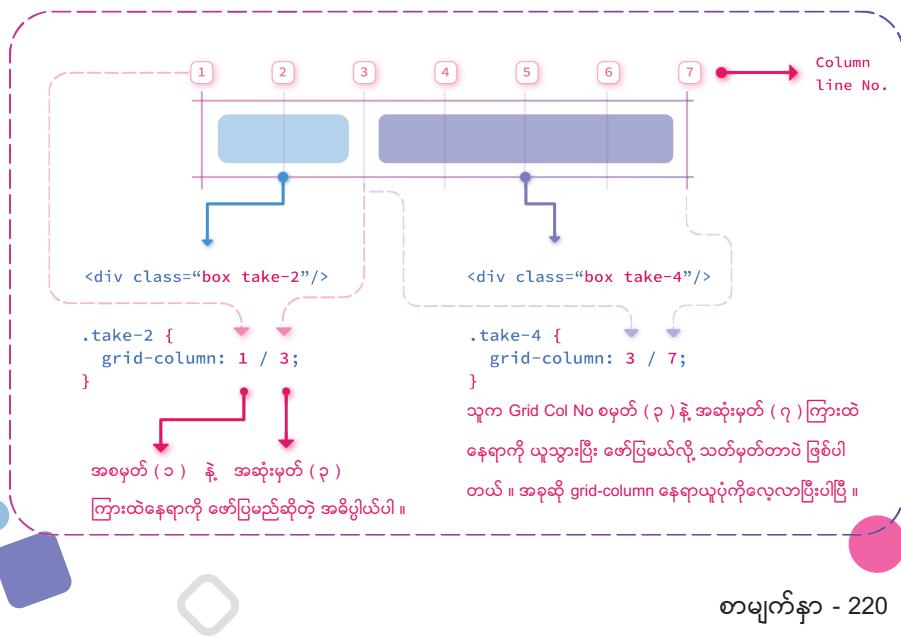
.take-4 {
    grid-column: 3 / 7;
}

.take-3-auto {
    grid-column: auto / span 3;
}
```

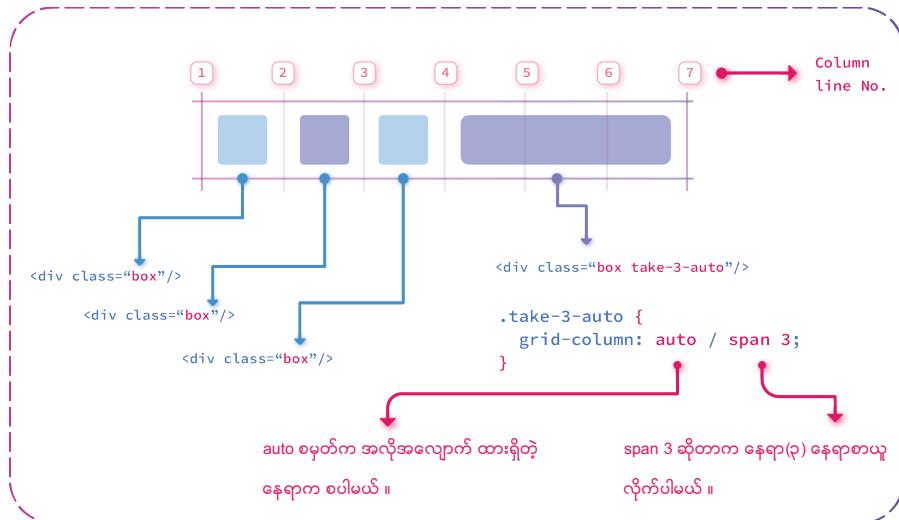
`grid-template-columns: repeat(6, 1fr);` ရေးသားထားတဲ့ အတွက် 1fr (၆) ခါလို တော့ Screen ရဲ့ အကျယ်ကို အောက်ကပုံအတိုင်း (၆) ပိုင်းပိုင်းလိုက်ပါမယ်။



အပေါ် Column Line Number (၁) ကနေ (၇) အထိဖော်ပြထားတာကို သတိထားပေးပါ။ သူက Grid Column တွေကို နေရာခြေမယ့်အပိုင်းမှာ အရေးကြီးပါတယ်။ ဥပမာ ကျွန်တော်တို့ fr အားလုံးပေါင်းလိုက်ခြင်းက 6fr ဆိုရင် Screen မှာ (၆) ပိုင်းရှစ်အတွက် Grid Column Line က (၁) ကနေ (၇) အထိမျဉ်းပိုင်းလိုက်မှ အလယ်မှာအကွက် (၆) ကွက် ကိုရမှာဖြစ်ပါတယ်။ အောက်က ပုံကို လေ့လာကြည့်ပါမယ်။ `grid-column` ဆိုတာက ရုက်က **Column No** ရဲ့ အစနဲ့ အဆုံးကို / (slash) ကြားထဲကခံပြီး ရေးသားသတ်မှတ်ရပါတယ်။



စမှတ်နံပါတ်ကို သုံးစရာမလိုပဲနဲ့ ကိုယ်သတ်မှတ်လိုက်တဲ့ Element ရဲ့ နေရာကပဲစချင် တယ်အခါရယ် အဆုံးမှတ်ကိုလည်း သိစရာမလိုပဲ ကိုယ်လိုချင်သလောက် Col နေရာကို ယူချင်တဲ့အခါမှာ အောက်က အတိုင်းရေးသားလို့ရပါတယ်။



- Grid Alignment Items ချိန်ညိုခြင်း

စာဖတ်သူတို့က **index.html** တွင် ဒီအောက်က Code ကိုကူးယူပြီးစမ်းကြည့်လို့ရပါမယ်။

<https://gist.github.com/lwinmoepaing/2514e79ef99e323008cb9cac151e4909>

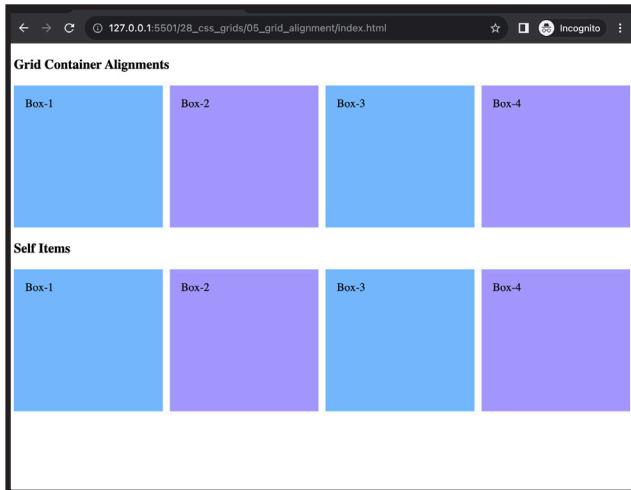
style.css တွင် ဒီအောက်က Code ကိုကူးယူပြီးစမ်းကြည့်ပါ။

<https://gist.github.com/lwinmoepaing/83b9885edb965ba379eeba9b9cc9dad2>

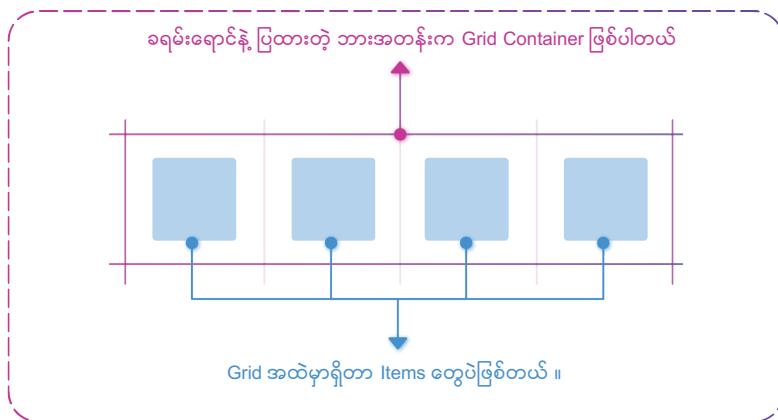
Codepen မှာလည်းတိုက်ရှိက် စမ်းကြည့်နိုင်ပါတယ်။

<https://codepen.io/lwinmoepaing/pen/XWoxOzR>

Browser ပေါ်မှာ Run လိုက်တဲ့အခါ အခုလို တွေ့ရမှုဖြစ်ပါတယ်။



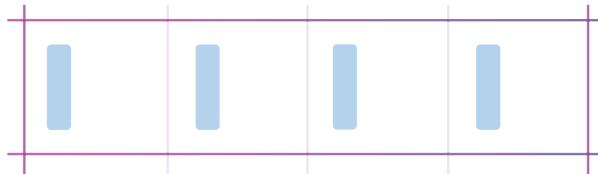
အရင်ဆုံး ကျွန်ုတ်တဲ့ Alignment ပြောင်းတာကို မလေ့လာခင်မှာ Grid Container ရဲ့ Grid Items ဆိုတာကို ရှင်းသွားအောင် အောက်ကပုံလေးကို တစ်ချက် လေ့လာပေးပါ။



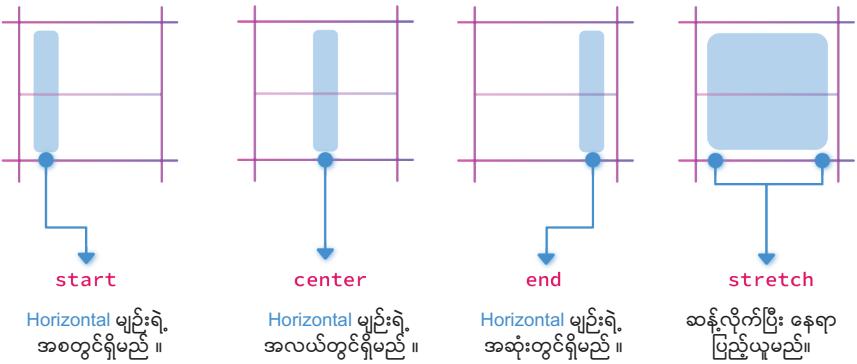
ပထမအနေနဲ့ Grid Container ကနေ အထဲမှာရှိတယ့် Grid Items တွေရဲ့ Position ကို ပြောင်းလဲသတ်မှတ်ပေးပုံကို အရင်လေ့လာပါမယ်။ ထူးခြားချက်အနေနဲ့က Container က သတ်မှတ်လိုက်တာနဲ့ Grid Items တိုင်းက သက်ရောက်မှုရှိသွားမှာပါ။

Grid Container ကို Select ထားတဲ့ .grid-1 ကို justify-items: start ဆိုပြီး CSS Rule သတ်မှတ်ပေးလိုက်ပါတယ်။ အဲဒါက အောက်ကုပ္ပါတီင်း သူ့အထဲမှာရှိတယ့် Items တစ်ခုချင်းတိုင်းကာဘယ်ဘက်ကို ကပ်သွားတယ့်ပုံဖြစ်သွားပါလိမ့်မယ်။

```
.grid-1 {
    justify-items: start;
}
```



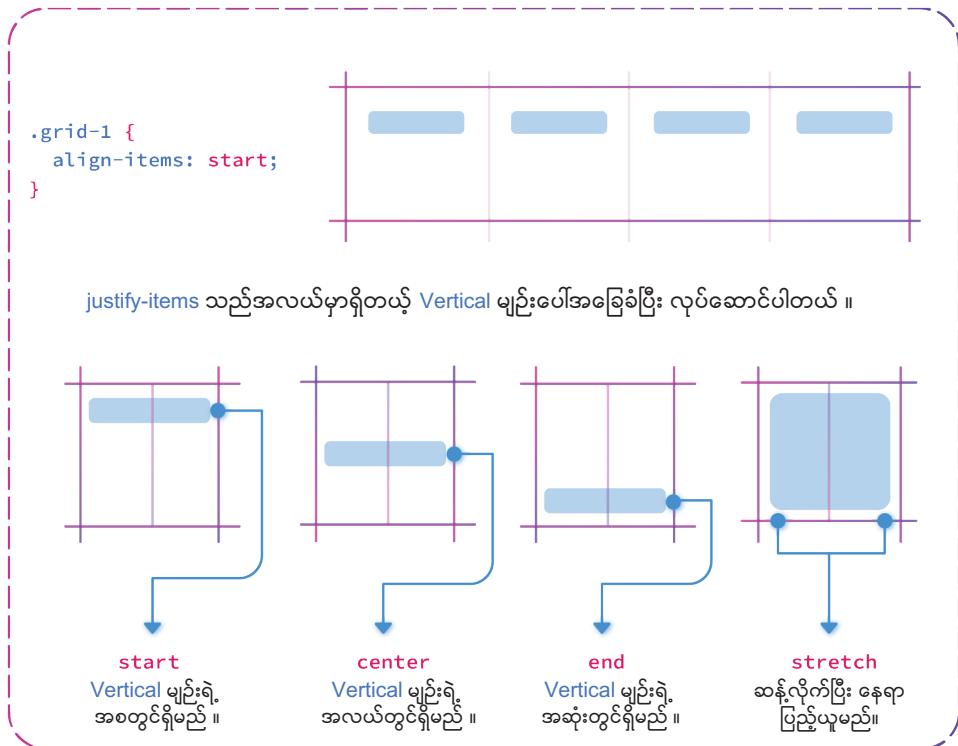
justify-items သည်အလယ်မှာရှိတယ့် Horizontal မျဉ်းပေါ်အခြေခံပြီး လုပ်ဆောင်ပါတယ်။



စာဖတ်သူတို့က Selector .grid-1 ကို justify-items: start ကနေ center, end, stretch ဆိုပြီး Browser ပေါ်တွင်ရေးသားစမ်းသပ်ကြည့်ပါ။ အတွင်းက Items တိုင်းက နေရာများပြောင်းလဲသွားတာကို တွေ့ရမှာဖြစ်ပါတယ်။

နေရာက်တစ်မျက်နှာမှာ justify-items မဟုတ်တော့ပနဲ့ Vertical မျဉ်းပေါ်က နေရာပေးတာကို လေ့လာသွားမှာဖြစ်ပါတယ်။

Grid Container ကို Select ထားတဲ့ .grid-1 ကို align-items: start ဆိုပြီး CSS Rule သတ်မှတ်ပေးလိုက်ပါတယ်။ အဲဒါက အောက်ကုပ္ပါတီင်း သူ့အထဲမှာရှိတယ့် Items တစ်ခုချင်းတိုင်းကအပေါ်ဘက်ကို ကပ်သွားမှာပါ။

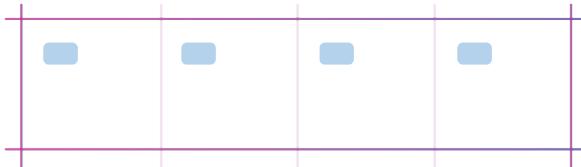


စာဖတ်သူတို့က Selector .grid-1 ကို align-items:start ကနေ center, end, stretch ဆိုပြီး Browser ပေါ်တွင်ရေးသားစမ်းသပ်ကြည့်ပါ။ အတွင်းက Items တိုင်းက နေရာများကြောင်းလွှားတာကို တွေ့ရမှာဖြစ်ပါတယ်။

နောက်တစ်မျက်နှာမှာတော့ နှစ်ခုလုံးပေါင်းစပ်ရေးတာတာကို လေ့လာကြပါမယ်။

Grid Container ကို Select ထားတဲ့ .grid-1 ကို align-items: start; justify-items: start ဆိုပြီး CSS Rule သတ်မှတ်ပေးလိုက်ရင် items တိုင်းက ဘယ်ဘက်အပေါ်ထောင့် ကို ရောက်သွားတာကို တွေ့ရှိရမှာပါ။ အောက်ကပုံမှာ လေ့လာကြည့်ရအောင်။

```
.grid-1 {
  justify-items: start;
  align-items: start;
}
```



align-items: start;
Vertical မျဉ်းရဲ့
အစတွင်ရှိမည်။

justify-items: start;
Horizontal မျဉ်းရဲ့
အစတွင်ရှိမည်။

{
 align-items: start;
 justify-items: start;
}
နှစ်ခုလုံးကို ထပ်လိုက်ပြီး ဘုံးတူစုံတဲ့
နေရာမှာ တည်ရှိပါမည်။

Browser ပေါ်တွင်
ပေါ်မည့် နေရာ

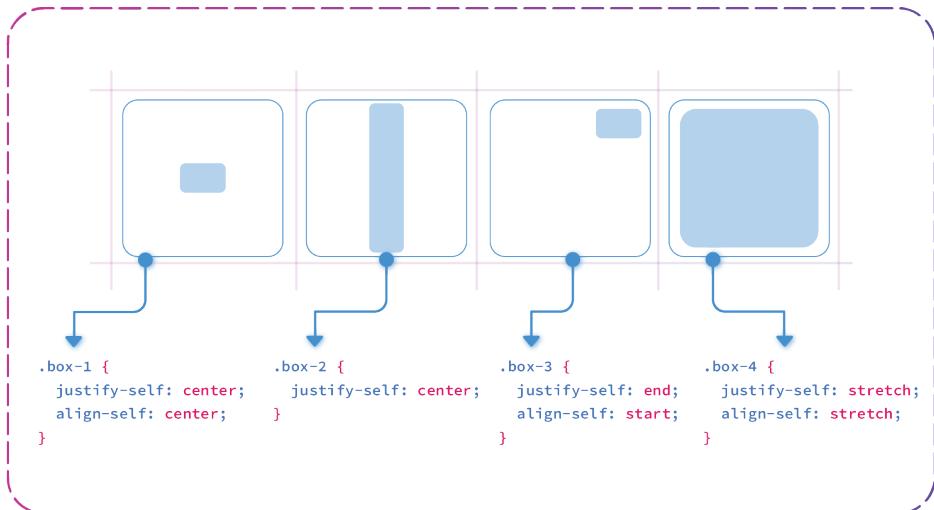
align-items: center; justify-items: center;
Vertical မျဉ်းရဲ့
အလယ်တွင်ရှိမည်။

justify-items: center;
Horizontal မျဉ်းရဲ့
အလယ်တွင်ရှိမည်။

{
 align-items: center;
 justify-items: center;
}
နှစ်ခုလုံးကို ထပ်လိုက်ပြီး ဘုံးတူစုံတဲ့
နေရာမှာ တည်ရှိပါမည်။

Browser ပေါ်တွင်
ပေါ်မည့် နေရာ

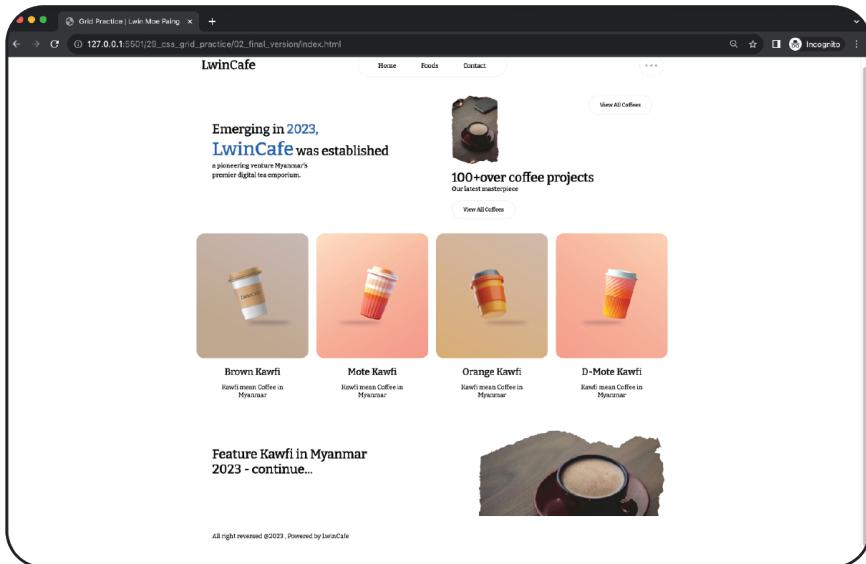
Grid Container မဟုတ်တော့ဘဲ အထဲမှာရှိတယ့် **Items** တစ်ခုချင်းဆီကိုယ်တိုင် **CSS Rule** တွေပေးပြီး ကြောင်းလဲသတ်မှတ်မှာဖြစ်ပါတယ်။ အောက်ကူပုံမှာ **.box-1**, **.box-2**, **.box-3**, **.box-4** ဆိုပြီး တစ်ခုချင်းဆီကို **Selector** အနေနဲ့တည်ရှိနေမှာပါ။



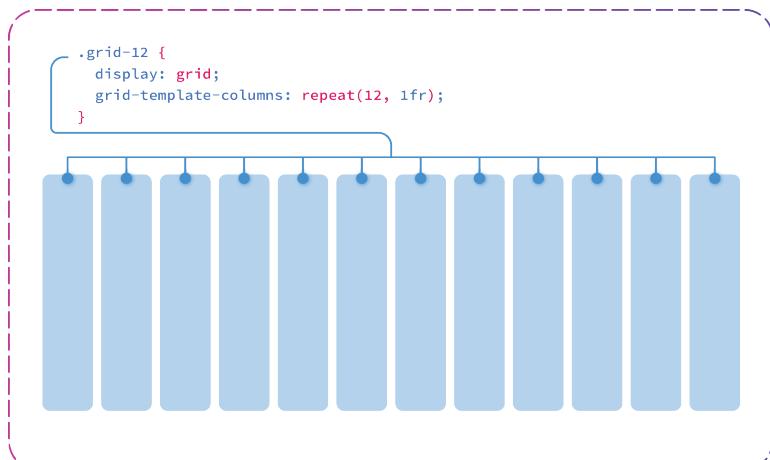
Items တစ်ခုချင်း **CSS Rule** သတ်မှတ်တဲ့အခါ **self** ကိုသတိထားပေးပါ။ **justify-self**, **align-self** ဆိုပြီး ကြောင်းလဲရေးသားပါတယ်။ အခုန်ရင် **Grid** အကြောင်းကို အခြေခံအားဖြင့် နားလည်သဘောက်ပေါက်ပြီလို့ ယူဆပါတယ်။ နောက်တစန်းမှာ ဘယ်လို အသုံးချက်မလဲ ဆိုတာတို့ လေ့လာသွားကြပါမယ်လျှော့။

Grid System ကိုအသုံးပြုခြင်း

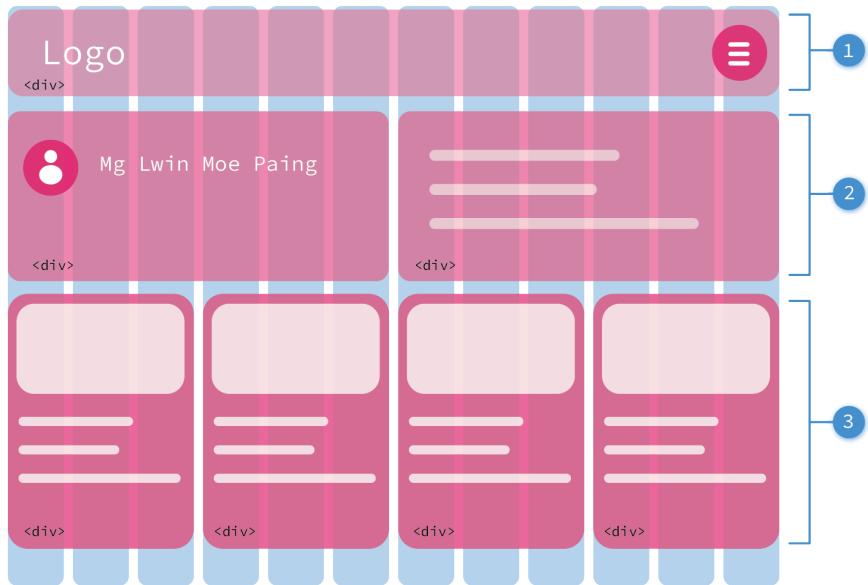
Grid System အသုံးပြုပုံကို ကျန်တော်က အောက်မှာပြထားတဲ့ Sample Website လေး
ကိုဖြည့်ပြီး ဘယ်လိုပုံစံနဲ့ အသုံးချပြီး ရေးဆွဲမလဲ လေ့လာကြမှာဖြစ်ပါတယ်။



ကျန်တော်တို့ အသုံးအများဆုံး ဖြစ်တဲ့ **12 Grids System** ကိုလေ့လာပါမယ်။ သူက ဘယ်လိုလဲဆိုရင် Web Designer အများစုက စံထားပြီး 12 columns ကို အခြေခြား
အသုံးပြုကြတို့ဖြစ်ပါတယ်။



အရင်ဆုံးပုံထဲမှာ ဖော်ပြထားတဲ့အတိုင်း **Column 12** ခုရှိတဲ့ Grid တစ်ခုကို
တည်ဆောက်လိုက်ပါမယ်။ Grid အကြောင်းကို စာဖတ်သူတို့က လေ့လာခဲ့ပြီဖြစ်တဲ့
အတွက် **.grid-12 class** ဖွင့်သတ်မှတ်ထားသော **CSS Rule** ကိန္ဒားလည်မယ်လို့
ယူဆပါတယ်။ အောက်က ပုံထဲကအတိုင်း **Row** (၃) ခုရှိပြီး ညာဘက်မှာ အညွှန်း 1 to 3
အထိ ဖော်ပြထားပါတယ်။



1.

```
.col-12 {  
    grid-column: auto / span 12;  
}
```

ပထမ Row အတွက် col-12 ဆိုပြီးကြော်လားခြင်းသည်
Grid-column 12 ခုအတိုင်းလည်မယ်ဆိုပြီး သတ်မှတ်ထား
တာဖြစ်ပါတယ်။

2.

```
.col-6 {  
    grid-column: auto / span 6;  
}
```

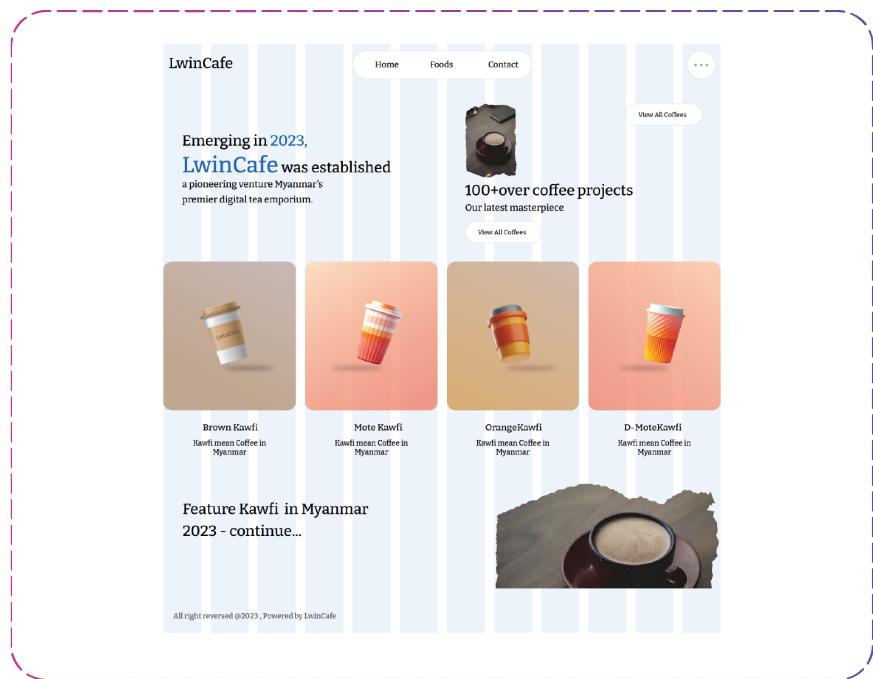
ဒုတိယ Row အတွက် ဝေ-6 ဆိုပြီးကြော်လားခြင်းသည်
Container တစ်ခုအတွက် grid-col 6 ခုအားလုံး
မှုဖြစ်ပါတယ်။ ဒေါ်အတွက် Screen ရဲ့ တစ်ဖက်အကျဉ်း
တစ်ဝက်ဆီရသွားမှာပါ။

3.

```
.col-3 {  
    grid-column: auto / span 3;  
}
```

တတိယ Row အတွက် ဝေ-3 ဆိုပြီးကြော်လား
ခြင်းသည် Container တစ်ခုအတွက် grid-col 3
ခုအားလုံးမှုဖြစ်ပါတယ်။ $3 \times 4 = 12$ ဖြစ်တဲ့
အတွက် 4 ပိုင်းအထိ အညီအမျှ ယူသွားမှာပါ။

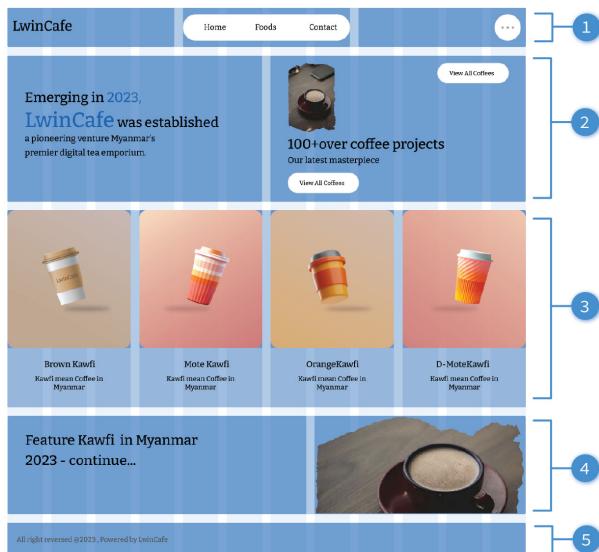
ဒါနိုင် လွှဲလာသွားမယ် Web Design ကို Col12 Grid နဲ့ ထပ်တည့်လိုက်ပါမယ်။



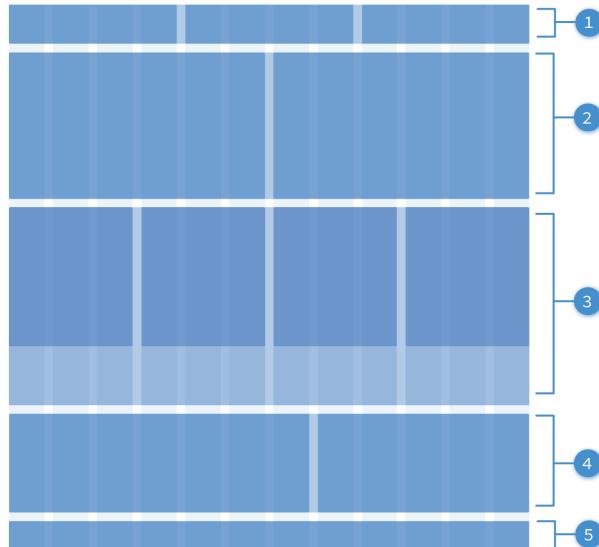
စာဖတ်သူတို့ မျက်လုံးထည့်မြင်သွားအောင် အောက်ကပုံအတိုင်း Rows တစ်ခုစီထပ်ခွဲလိုက်ပါမယ်။



အခုက္ခန်တော်တို့ **Columns** ဘယ်လောက်ဆီယူမလဲ မြင်သွားအောင် နောက်ထပ်ပံ့ကို ကြည့်လိုက်ရင်၊ ပထမ လေ့လာခဲ့သလို စာဖတ်သူတို့မျက်လုံးထဲမြင်သွားနိုင်ပါပြီ။



ဒီထက်ပိုဂျင်းသွားအောင် ကျွန်ုတ်တို့ ပုံတွေရယ် စာတွေကိုခက်ဖျောက်လိုက်ပါမယ်။



ဆိုလိုချင်တာက **Layout** ကိုချိန်ပြီဆိုတာနဲ့ ကျွန်တော်တို့က Website ရဲ့ 30% လောက်ပြီးမြောက်သွားပြီလို့ အကြပ်းအားဖြောဆိုနိုင်ပါတယ်။ နောက်ဆုံးဖော်ပြထားတဲ့ ပုံအတိုင်း **Layout** ကို စာဖတ်သူတို့က ချမှတ်နိုင်စေရန် **Code** တွေကို လေ့လာကြည့်ပါမယ်။

စာဖတ်သူတို့က **index.html** တွင် ဒီအောက်က Code ကိုကူးယူပြီးစမ်းကြည့်လို့ရပါမယ်။

<https://gist.github.com/lwinmoepaing/ba5c8500a7322798c328b3c158697950>

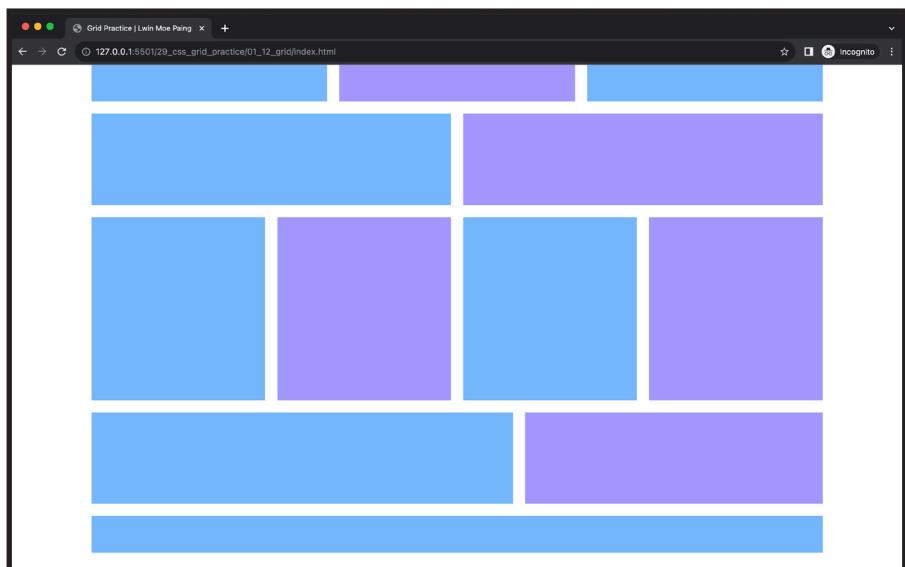
style.css တွင် ဒီအောက်က Code ကိုကူးယူပြီးစမ်းကြည့်ပါ။

<https://gist.github.com/lwinmoepaing/3f7ddcdfca76d56a9201147272c3234f>

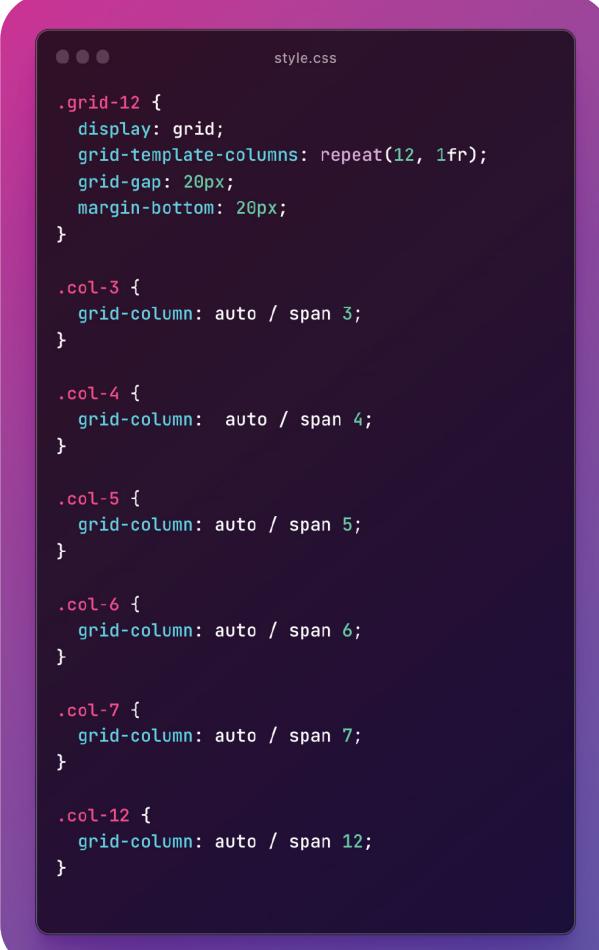
Codepen မှာလည်းတိုက်ရှိက် စမ်းကြည့်နိုင်ပါတယ်။

<https://codepen.io/lwinmoepaing/pen/wvNaQvO>

စာဖတ်သူတို့က Browser ပေါ်တွင် စမ်းလိုက်တဲ့အပါ အောက်က ပုံအတိုင်းပေါ်လာပါမည်။



style.css အတွင်းရေးသားထားတဲ့ပုံများကို လေ့လာကြည့်ရအောင် ... !!



```

style.css

.grid-12 {
    display: grid;
    grid-template-columns: repeat(12, 1fr);
    grid-gap: 20px;
    margin-bottom: 20px;
}

.col-3 {
    grid-column: auto / span 3;
}

.col-4 {
    grid-column: auto / span 4;
}

.col-5 {
    grid-column: auto / span 5;
}

.col-6 {
    grid-column: auto / span 6;
}

.col-7 {
    grid-column: auto / span 7;
}

.col-12 {
    grid-column: auto / span 12;
}

```

အရင်ဆုံး style.css ထဲမှာ ဖော်ပြထားတဲ့အတိုင်း .grid-12 class ဖြင့် Column 12 ခု ရှိတဲ့ Grid တစ်ခုကို တည်ဆောက်လိုက်ပါမယ် !! သူက row တစ်ခုခြင်းယူတဲ့အပါမှာ Col 12 ခုဖြင့် Layout ချုပ်မယ့်အပါတိုင်း သုံးမှာပါ !! အောက်က .col နဲ့ စတဲ့ class တွေ ကတော့ column ဘယ်လောက်ထိယူမှာဖြစ်ကြောင်း သတ်မှတ်ထားတာဖြစ်ပါတယ် !!

```
index.html

<div class="grid-12">
  <div class="box col-4 min-h-60px"></div>
  <div class="box col-4 min-h-60px"></div>
  <div class="box col-4 min-h-60px"></div>
</div>
<div class="grid-12">
  <div class="box col-6 min-h-150px"></div>
  <div class="box col-6 min-h-150px"></div>
</div>
<div class="grid-12">
  <div class="box col-3 min-h-300px"></div>
  <div class="box col-3 min-h-300px"></div>
  <div class="box col-3 min-h-300px"></div>
  <div class="box col-3 min-h-300px"></div>
</div>
<div class="grid-12">
  <div class="box col-7 min-h-150px"></div>
  <div class="box col-5 min-h-150px"></div>
</div>
<div class="footer">
  <div class="box min-h-60px"></div>
</div>
```

`index.html` အတွင်းမှာတော့ `.grid-12` နဲ့ container တစ်ခုပြုလုပ်မယ်။ အတွင်းထဲရှိ `child element` တွေကတော့ `col-4` ဆိုရင် `Column` အရေအတွက် 4 နေရာဆီယူသွားမှာဖြစ်ပြောတော့ အစရိတ်ဖြင့် သတ်မှတ်ထားတာဖြစ်ပါတယ်။ ဒါက စာဖတ်သူတို့ကိုယ်တိုင် Code ရေးလိုက်တဲ့အခါမှာ ပိုပြီးမြင်သွားမှာပါ။

ထောက Row အတွင်းမှာ `col-4` နဲ့ child element 3 ခုရှိပါတယ်။ $4\text{col} \times 3 = 12$ ဖြစ်တဲ့ အတွက် 3 ပိုင်းအညီအမျှပိုင်းသွားမှာပါ။ ဒုတိယ Row ကတော့ $6+6 = 12$ ဖြစ်တဲ့ အတွက် Screen ကို J ပိုင်း အညီအမျှယူသွားမှာဖြစ်ပါတယ်။ တတိယ Row ကတော့ $3\text{col} \times 4 = 12$ ဖြစ်တာဖြောင့် 4 ပိုင်းခွဲသွားမှာဖြစ်ပါတယ်။ နောက် Row က $7+5 = 12$ ဖြစ်ပြီး၊ Footer ကတော့ အပြည့်မို့လို့ Grid နဲ့မယူပဲ `<div>` element တစ်ခု အသုံးပြုလိုက်ပါတယ်။

အချလောက်အထိကို စာဖတ်သူတို့ က Grid အခန်းမှာသေသေချေချာ လေ့လာထားခဲ့ပြီး ဆိုရင် လွှာယ်လွှာယ်ကုကုပဲ နားလည်းသဘောပေါက်နှင့်မယ်လို့ ယူဆပါတယ်။ စာဖတ်သူတို့က အခြားအလာတူ Layout ပုံစံ တွေကိုလည်း ကိုယ်တိုင်ရေးဆွဲနိုင်သွားပါမည်။

ကျွန်တော်တို့ ပြုလုပ်မယ့် Web Design ကုတ်တွေမှာ စာဖတ်သူတို့ ရေးသားခဲ့တဲ့ Grid Layout ကိုဘယ်လိုအသုံးပြုထားမလဲဆိတာ ကို မြင်သွားအောင်၊ Web Design အတွက် ရေးထားတဲ့ Code တွေမှာ သွားရောက်လေ့လာကြပါမယ်။

Final Code Files တွေအတွက် ဒီအောက်က Drive Link မှာ အားလုံးခေါင်းလှစ် လုပ်ရန်ကိုနှိပ်ပါ ပြီးရင် ကိုယ့်ရဲ့ စက်ထဲမှာ index.html ကိုဖွင့်လိုက်ပါ။

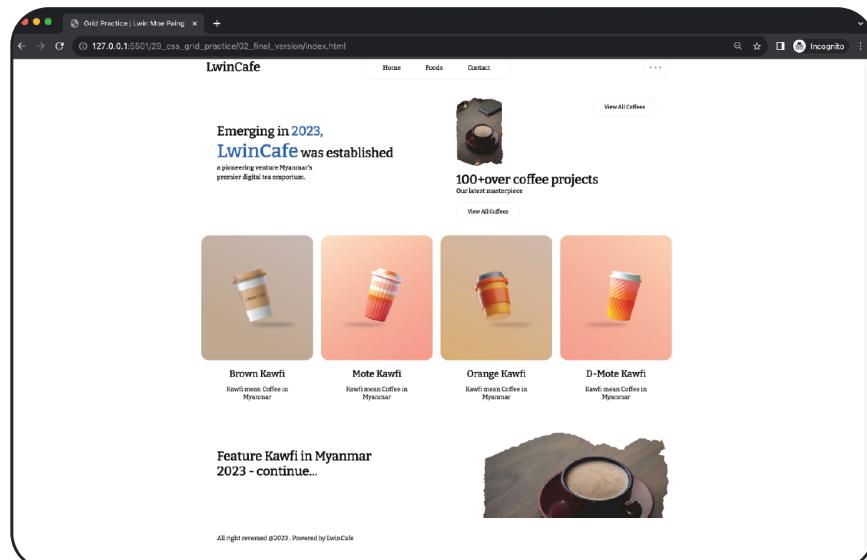
<https://drive.google.com/drive/folders/1vosJssFAgo-niT-LIsDAoa4B8bCtlYHd?usp=sharing>



(သို့မဟုတ်) Codepen မှာလည်းတိုက်ရိုက် စမ်းကြည့်နိုင်ပါတယ်။

<https://codepen.io/lwinmoeapaing/pen/bGOEJVK>

Browser ပေါ်တွင်အောက်ကပုံအတိုင်း LwinCafe ဆိုတဲ့ Website လေးကို မြင်ရမှာဖြစ်ပါတယ်။



```
index.html
<link rel="stylesheet" href="css/style.css" />
<link rel="stylesheet" href="css/grid.css" />
<link rel="stylesheet" href="css/cafe.css" />
```

အရင်ဆုံး CSS ထဲမှာဖိုင် (၃) ဖိုင်ကို ချိတ်ထားတာတွေရမှာပါ။ ပထမဖိုင်ထဲမှာက Helper Style အကူတွေရေးထားတာဖြစ်ပြီး၊ grid.css ထဲမှာက စာဖတ်သူတို့ လေ့လာခဲ့တယ့် Column နဲ့ Grid တွေအကြောင်းရေးထားတာပါ။ cafe.css ထဲမှာတော့ Lwin-Cafe အတွက်သက်ဆိုင်သော Style တွေကို ရေးထားတာဖြစ်ပါတယ်။

```
index.html
<!-- Navigation Bar: Start -->
<div class="grid-12 navbar">
    <div class="col-4">
        <h2 class="nav-title font-title font-bold">LwinCafe</h2>
    </div>
    <div class="col-4">
        <div class="nav-link-group item-around-helper">
            <a href="#" class="link font-bold color-black font-body">Home</a>
            <a href="#" class="link font-bold color-black font-body">Foods</a>
            <a href="#" class="link font-bold color-black font-body">Contact</a>
        </div>
    </div>
    <div class="col-4">
        <div class="menu-icon item-center-helper">
            
        </div>
    </div>
</div>
<!-- Navigation Bar: Finish -->
```

grid-12 ဆိုတဲ့ Parent grid container နဲ့ အတွင်းမှာပဲ col-4 နဲ့ (၃) ပိုင်းခွဲထားပါတယ်။ 4x3=12 ဖြစ်တဲ့အတွက် အညီအမျှဆီ (၃) ပိုင်းရသွားမှာပါ။ အထဲမှာ Navigation အတွက် လိုအပ်တဲ့ style တွေကို .nav-title, .nav-link-group အစရိတဲ့ class rules တွေကို cafe.css မှာ စာဖတ်သူတို့က ဝင်ရောက်လေ့လာနိုင်ပါတယ်။ အကြွော်တွေ နားလည်နေပြုဖို့အတွက် လွယ်လွယ်ကူကူပဲ နားလည်လိမ့်မယ်လို့ မျှော်လင့်ပါတယ်။

```

    ● ● ●
    index.html

<!-- Section: Start -->
<div class="grid-12 section">
    <!-- Section Left-Side: Start -->
    <div class="col-6">
        <div class="section-title-group">
            <h2 class="font-bold font-title">
                Emerging in <span class="color-blue">2023,</span>
            </h2>
            <h2 class="font-bold font-title">
                <span class="font-large-title color-blue">LwinCafe</span> was
                established
            </h2>
            <h2 class="font-bold font-title font-body">
                a pioneering venture Myanmar's
            </h2>
            <h2 class="font-bold font-title font-body">
                premier digital tea emporium.
            </h2>
        </div>
    </div>
    <!-- Section Left-Side: Finish -->
    <!-- Section Right-Side: Start -->
    <div class="col-6">
        <div class="section-right-container">
            <div class="margin-bottom-16px item-between-helper">
                
                <button class="button">View All Coffees</button>
            </div>
            <div class="section-right-text">
                <h2 class="font-bold font-title">100+ over coffee projects</h2>
                <h2 class="font-bold font-body margin-bottom-20px">
                    Our latest masterpiece
                </h2>
                <button class="button">View All Coffees</button>
            </div>
        </div>
    </div>
    <!-- Section Right-Side: Start -->
</div>
<!-- Section: Finish -->

```

ပထမဆုံးမြင်ရမယ့် **Section** အတွက် **Column** (၂) ပိုင်းခွဲချင်တဲ့အတွက် **col-6** ကိုသုံး
ထားတာကို တွေ့မြင်ရမှာပါ။ အတွင်း **Rules** တွေကတော့ **cafe.css** နဲ့ **styles.css**
ထဲက **class** တွေကိုပါသုံးဘွားတဲ့အတွက် စာဖတ်သူတို့က ဖတ်လိုက်တာနဲ့ ဘယ်လို သုံး
ဘွားတာလဲဆိုတာကို အလွယ်တကူသိရှိနိုင်ပါတယ်။

```
index.html

<!-- Coffee List: Start -->
<div class="grid-12 coffee-lists">
    <div class="col-3">
        <div class="coffee-card-container">
            <div class="coffee-card-img coffee-first-card-img"></div>
            <div class="coffee-card-text-container">
                <h2 class="card-title">Brown Kawfi</h2>
                <p class="font-body margin-bottom-20px">
                    Kawfi mean Coffee in Myanmar
                </p>
            </div>
        </div>
    <div class="col-3">
        <div class="coffee-card-container">
            <div class="coffee-card-img coffee-second-card-img"></div>
            <div class="coffee-card-text-container">
                <h2 class="card-title">Mote Kawfi</h2>
                <p class="font-body margin-bottom-20px">
                    Kawfi mean Coffee in Myanmar
                </p>
            </div>
        </div>
    </div>
    <div class="col-3">
        <div class="coffee-card-container">
            <div class="coffee-card-img coffee-third-card-img"></div>
            <div class="coffee-card-text-container">
                <h2 class="card-title">Orange Kawfi</h2>
                <p class="font-body margin-bottom-20px">
                    Kawfi mean Coffee in Myanmar
                </p>
            </div>
        </div>
    </div>
    <div class="col-3">
        <div class="coffee-card-container">
            <div class="coffee-card-img coffee-fourth-card-img"></div>
            <div class="coffee-card-text-container">
                <h2 class="card-title">D-Mote Kawfi</h2>
                <p class="font-body margin-bottom-20px">
                    Kawfi mean Coffee in Myanmar
                </p>
            </div>
        </div>
    </div>
<!-- Coffee List: Finish -->
```

ကော်မိကမ်လေးတွေအတွက် **Section** မှတေသူ **Column** (ငဲ) ပိုင်းခွဲပြီး ဖော်ပြထားတာ ကိုမြင်ရမှာပါ။ **col-3** ကိုသုံးထားပြီး အထဲမှတေသူ **Card** တွေအတွက် **class** တွေထုံးထားတာကို မြင်ရမှာဖြစ်ပါတယ်။ ဒီလောက်ဆိုရင် ကျွန်တဲ့ Styles တွေလည်း လွယ်သွားလိမ့်မယ်လို့ ထင်ပါတယ်။ နောက်အခန်းမှာ Responsive အကြောင်းအရာတွေကို လေ့လာကြည့်ရအောင် ...။

Responsive CSS ကိုလေ့လာခြင်း

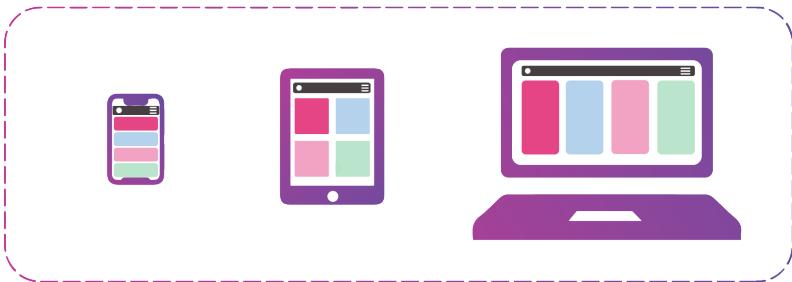
Responsive ဆိုတာဘာကြီးလဲ ဘာကြောင့်လေ့လာသင့်?

အခါ မော်ဒန်ခေတ်မှာ ကျွန်တော်တို့ Website တစ်ခုကို နေရာအရုံ Devices အစုံက အသုံးပြုလာကြပါပြီ။ မျက်လုံးထဲမြင်အောင်ပြောရရင် Mobiles, Tablets, Laptops က စပိုး Large Screen Monitor တွေ Smart TV တွေကအစ လုမ်းပြီးအသုံးပြုနိုင်နေကြပါဖြစ်ပါတယ်။ အဆိုးဆုံးကတော့ Smart watch ဖြစ်တဲ့ လက်ပတ်နာရီလေးတွေမှာ တောင် Browser အသုံးပြုလို့ရနိုင်တဲ့အတွက် website တွေကို access ပြုလုပ်လို့ရပါတယ်။

ဒီလို မော်ဒန် Devices တွေကြောင့် စာဖတ်သူတို့၊ ကျွန်တော်တို့ ကိုယ်တိုင်ကလည်း ရင်ဆိုင်ရမယ့် အကြောင်းအရာ တစ်ခု ရှိလာပါတယ်။ ဘာလဲဆိုရင် Mobile ဖုန်းတွေရဲ့ Screen Size ရယ်၊ Smart TV တို့ရယ် Screen Size အကျယ်တွေက မတူညီကြတာ ပါပဲ။ ဒီအတွက်ကြောင့် ဘယ် Screen Size အကျယ်မှာမဆို ကျွန်တော်တို့ ရေးဆွဲထားတဲ့ Website ရဲ့ပုံစံက ပျက်ပျဉ်း [ယွေးတောင်း] တာမျိုးတွေမရှိအောင်လုပ်ဖို့က အရေးကြီးလာပါတယ်။ ဘယ် Device ရဲ့ Screen မှာမဆို အဆင်ပြေအောင် Responsive ဖြစ်အောင် ရေးသားနိုင်ရမယ့် တာဝန်က Web developer အတွက် တာဝန်တစ်ခု ဖြစ်လာပါတယ်။

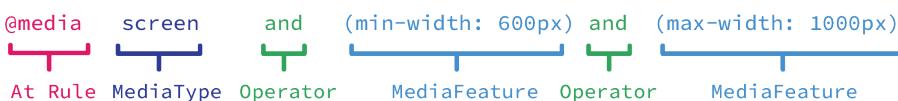


အခါဖော်ပြပါ ပုံထဲမှာ Laptop အတွက် Column (ငါ) ခုစာယူထားတဲ့ Layout ကိုမြင်ရမှာပါ။ Tablet မှာလဲ (ငါ) ပိုင်း အနေနဲ့ပဲ ဖော်ပြထားပါတယ်။ Mobile အတွက်လည်း အတူတူပါပဲ။ စဉ်းစားကြည့်ရအောင် အထဲက Card တစ်ခုချင်းဆီမှာ စာတွေပါနေမယ်။ ပုံ Content တွေပါနေပါလိမ့်မယ်။ တကယ့် Mobile နဲ့ Tablet Screen ကိုရောက်သွားရင် ထိုစာတွေ ပုံတွေက ပူးကပ်ပြီး ဖတ်ရတာအဆင်မပြတဲ့ပုံစံမျိုးနဲ့ ဖော်ပြသွားမှာပါ။ အသုံးပြုသူတွေအတွက်တော့ အဆင်ပြတော့မှာမဟုတ်ပါဘူး။



ဒီ Layout မှတော့ ဖြောင်းလဲသွားပါပြီ။ Laptop Screen မှာ Column (၄) ရနဲ့ တစ်ပုံစံ၊ Tablet Screen အတွက် Column (၂) ရနဲ့ Card (၄) ခုကို စွဲပြတဲ့ ပုံစံရယ်၊ ဖုန်းစခရင် အတွက်ကတော့ အပြည့်ယူလိုက်တဲ့ Card ပုံစံလေးကိုတွေ့ရမှာပါ။ ဒါဆိုရင် စာတွေ ပုံတွေက ဘယ် Screen Size မှာဖြစ်ဖြစ် အဆင်ပြေသွားမှာဖြစ်ပါတယ်။ နောက်တစ်ပိုင်းမှာ Responsive CSS ရေးသားနည်းကို လေ့လာသွားကြပါမယ်။

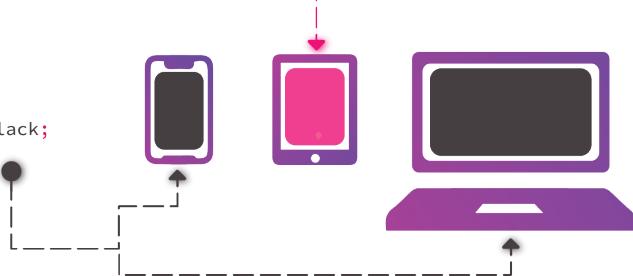
Responsive CSS ရေးထုံး ရေးသားနည်း



လေ့လာခဲ့ပြီးတဲ့အတိုင်း `@` နဲ့စတဲ့ **CSS** ရေးသားနည်းကို **Font** အခန်းမှာလည်းတွေ့ပြီး ကြပါဖော်ပါတယ်။ **At Rule** အဖြစ် `@media` နဲ့စပြီး ရေးသားရပါတယ်။ ပြီးရင် **Media Type** ကို ကြော်ပေးရပါတယ်။ ဒီမှတော့ **screen** လိုသုံးထားပါတယ် **screen** တွေ အတွက်ပဲ **Responsive** ရေးသားမှာမို့လို့ပါ။ ဒီနေရာမှာ **print** ဆိုတဲ့ **type** ကိုလည်း အသုံးပြုလိုရပါတယ် သူကတော့ **Print** ထုတ်တဲ့အခါမျိုး ဥပမာ **PDF** ဖိုင်ထုတ်တာ မျိုးတွေမှာဆိုရင် **Apply** လုပ်ချင်တဲ့အခါသုံးလို့ရပါတယ်။ ပြီးရင် **Operator** ရေးပေးရပါမယ် **and** ကိုသုံးပါတယ် **or** အတွက်လည်း **(,)** ကို အသုံးပြုပေးမယ် ဒီမှတော့ ကျွန်ုတ် အဲအကြောင်းကိုဖော်ပြသွားမှာမဟုတ်ပါဘူး။

နောက် **Media Feature** ကိုရေးပေးရမှာဖြစ်ပါတယ် ဘယ် **Screen Size** အတွက်လည်း ဆိုတာ သတ်မှတ်ပေးရတာပါ။ ပြီးရင် နောက်ထပ် ထပ်ပြီး **Screen Size** တွေအတွက် ထပ်သတ်မှတ်ချင်သေးပါက **Operator** နဲ့ **Media Feature** ဆိုပြီး ရေးသားနိုင်ပါတယ်။ ဒီနေရာမှာ စာဖတ်သူတို့က အဂွတ်ရှိနိုးမလိုပါဘူး ရေးသားနည်းကို နားလည်သွားရင်ရ ပါတယ်။ နောက်အပိုင်းတွေမှာ ရေပါများသွားရင် အလိုလို ရသွားပါလိမ့်မယ်။

```
@media only screen and (min-width: 600px) and (max-width: 1000px) {
  .container {
    background-color: red;
  }
}
.container {
  background-color: black;
}
```



အခုရေးသားနည်းမှာ Container ကို Background Color အမည်းအနေနဲ့ သတ်မှတ်ထားတာဖြစ်ပါတယ်။ သို့ပေါ့မယ် @media ရေးတုံးရေးထဲ Screen တွေအတွက် ကျွန်ုတ်တို့က (min-width: 600px) အဓိပ္ပာယ်က အနည်းဆုံး 600px ရှိတဲ့ Screen ကနေ and operator နဲ့ဆက်ပြီး (max-width: 1000px) အများဆုံး 1000px ကြားရှိတဲ့ Screen တွေအတွက်ဆိုပြီး ... အတွင်းမှာ CSS Rule အဖြစ် Container ရဲ့ Background Color ကို အနီအဖြစ် ထားထားပေးပါဆိုပြီး သတ်မှတ်ထားလိုက်ပါတယ်။ ဒီအတွက် ကြောင့် Laptop နဲ့ Mobile ဖုန်းမှာတော့ အနက်ရောင်အဖြစ်မြင်ရမှာဖြစ်ပြီး Tablet တွေ ရဲ့ Screen မှာတော့ အနီရောင်အဖြစ်မြင်ရမှာပါ။

စာဖတ်သူတို့က index.html တွင် ဒီအောက်က Code ကိုကူးယူပြီးစမ်းကြည့်လို့ရပါမယ်။

<https://gist.github.com/lwinmoepaing/54fd7acad8635185c3ffc24d49efbefd>

style.css တွင် ဒီအောက်က Code ကိုကူးယူပြီးစမ်းကြည့်ပါ။

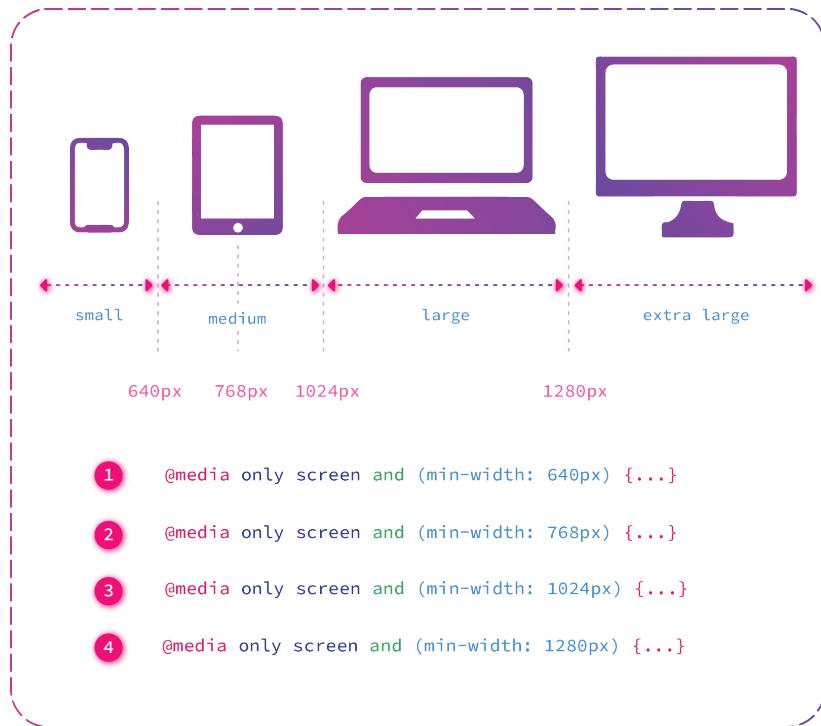
<https://gist.github.com/lwinmoepaing/9601a1df9b1e638c8eba880097eb55c1>

Codepen မှာလည်းတိုက်ရှိက် စမ်းကြည့်နိုင်ပါတယ်။

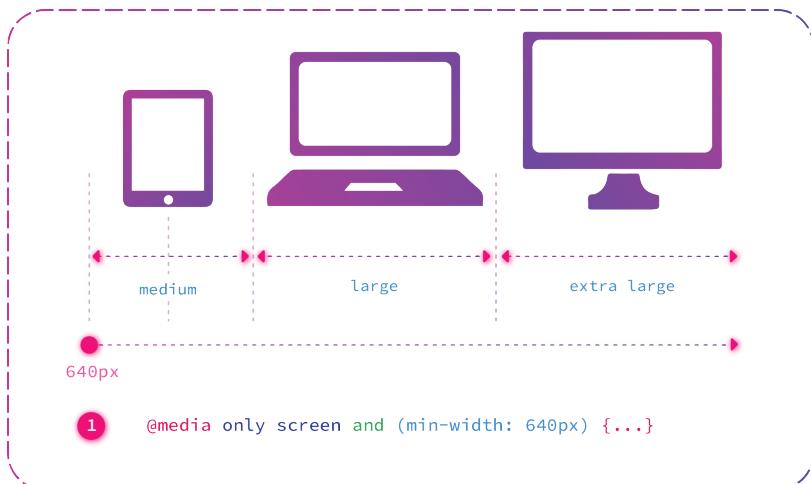
<https://codepen.io/lwinmoepaing/pen/Rwrvdp>

Responsive CSS Minimum Width ရေးသားနည်း

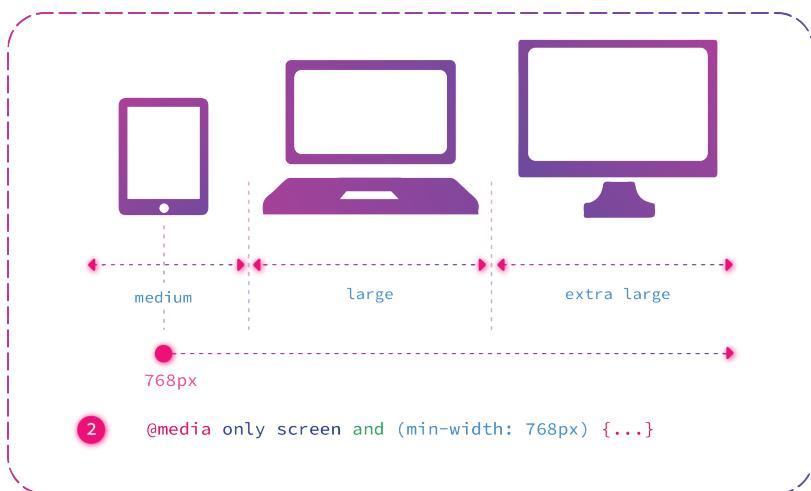
အခုရေးသားနည်းက `min-width` ရေးထံးပဲဖြစ်ပါတယ်။ အဓိကအားဖြင့် Responsive Column Grid တွေမှာ အသုံးအများဆုံးနည်းလည်းပဲဖြစ်ပါတယ်။ အရင်ဆုံး အောက်က ပုံလေးထဲက `Breakpoints` တွေကို လေ့လာကြပါမယ်။



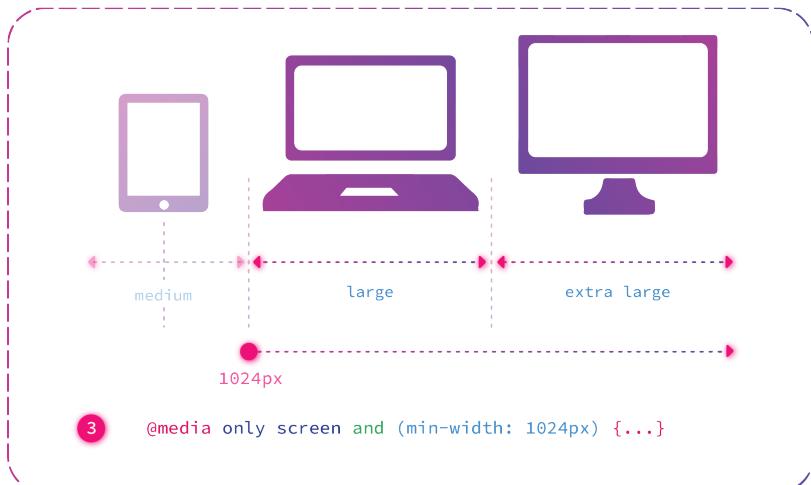
Tablets Size တွေအတွက် 640px ကနေ 1024px အတွင်းလို့ အကြမ်းအားဖြင့် သတ်မှတ်နိုင်ပါတယ်။ 1024px ကနေအထက်ကို Large Screen တွေအတွက် သတ်မှတ်ထားတာပါ။ သို့ပေါ်မယ် ဒီနံပါတ်တွေက အမြဲပံ့သေလို့ သတ်မှတ်လို့မရပါဘူး အပြင်လေက်တွေ၊ Project တွေမှာ Designer Team တွေက သတ်မှတ်ထားတဲ့ နံပါတ် တွေကို သုံးပြီး ဘယ် Screen အတွက်ဘယ်လောက် Pixels ကနေစမယ်ဆိုပြီး စံ သတ်မှတ်လေ့ရှိပါတယ်။ အခုပုံထဲက `Breakpoints` နံပါတ်တွေကတော့ အများအားဖြင့် သုံးလေ့ရှိကြတဲ့ Px နံပါတ်တွေကိုဖော်ပြထားပြီး လေ့လာသွားမှာဖြစ်ပါတယ်။



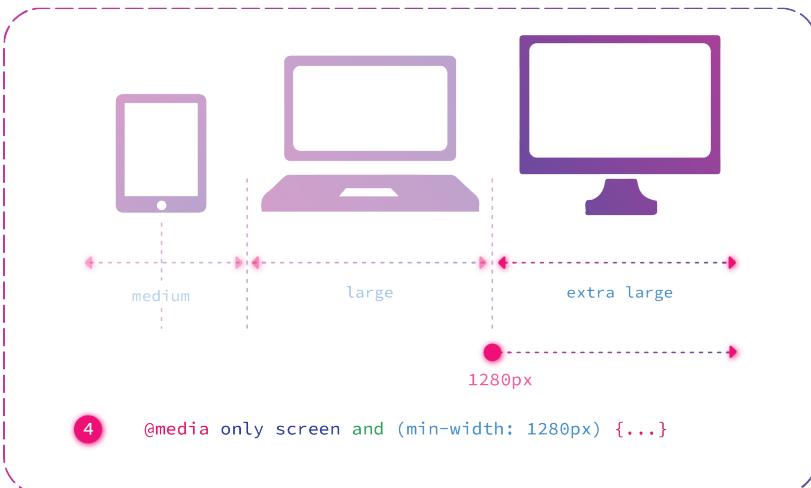
@media only screen ဆိတာက Screen တွေအတွက်ပလို့ သတ်မှတ်တာဖြစ်ပြီး (min-width: 640px) အနည်းဆုံး 640px ကနေစပြီးတော့ CSS Rule တွေသတ်မှတ်မယ်လို့ ကြော်ထားတာဖြစ်ပါတယ်။ Tablet Screen ကနေစပြီး နောက်ပိုင်း Large Screen တွေအတိ အကြံးဝင်ပါတယ်။



@media only screen ဆိတာက Screen တွေအတွက်ပလို့ သတ်မှတ်တာဖြစ်ပြီး (min-width: 768px) အနည်းဆုံး 768px ကနေစပြီးတော့ CSS Rule တွေသတ်မှတ်မယ်လို့ ကြော်ထားတာဖြစ်ပါတယ်။ Tablet Screen ထဲမှာမှာ ပိုပြီး Screen Size ကြီးတယ် Tablet တွေကစပြီး နောက်ပိုင်း Large Screen တွေအတိ ပါမယ်လို့ သတ်မှတ်ထားတာဖြစ်ပါတယ်။



(`min-width: 1024px`) အနည်းဆုံး `1024px` ကနေစပိုးတော့ CSS Rule တွေသတ်မှတ်မယ်လို့ ကြော်လော်ထားတာဖြစ်ပါတယ်။ Laptop Screen ကနေစပိုး နောက်ပိုင်း Large Screen တွေအထိ သတ်မှတ်လိုက်တာပါ။



(`min-width: 1280px`) အနည်းဆုံး `1280px` ကနေစပိုးတော့ CSS Rule တွေသတ်မှတ်မယ်လို့ ကြော်လော်ထားတာဖြစ်ပါတယ်။ Extra Large Screen ကနေစပိုး အကျိုးဝင်ပါတယ်။

စာဖတ်သူတို့က [index.html](#) တွင် ဒီအောက်က Code ကိုကူးယူပြီးစမ်းကြည့်လို့ရပါမယ်။

<https://gist.github.com/lwinmoepaing/bad5c9f69aeb450cd08b9d5f1812f2cd>

[style.css](#) တွင် ဒီအောက်က Code ကိုကူးယူပြီးစမ်းကြည့်ပါ။

<https://gist.github.com/lwinmoepaing/8ba749962d71b6fd6271f9cb5b188992>

Codepen မှာလည်းတိုက်ရှိက် စမ်းကြည့်နိုင်ပါတယ်။

<https://codepen.io/lwinmoepaing/pen/QWYyRQm>

အရင်ဆုံး Browser ပေါ်တွင် Run လိုက်တဲ့အခါ **Title Header** က စာလုံးအနီရောင် နဲ့အောက်မျဉ်းလိုင်းနဲ့ပါတာကိုတွေ့ရမှာပါ။ **Screen** ကို ချုပြုကြည့်လိုက်ရင်တော့ နောက်ဆုံး ဘယ်ဘက် **ဘေးအနီရောင်မျဉ်းလိုင်းနဲ့ Title Font Size** က လည်းသေးသွား တာကို စာဖတ်သူတို့ကတွေ့မြင်ရမှာဖြစ်ပါတယ်။

.title ကိုရေးထားတာကို Code တွေဝင်ဖတ်ကြည့်ရအောင် ...။ **border-left** အနီရောင်နဲ့ **font-size: 30px** ပေးထားတာကို စာဖတ်သူတို့တွေ့ရမှာဖြစ်ပါတယ်။ အဲဒါက Default Value တန်ဖိုးအနေနဲ့ရှိမှာပါ အဲအတွက် Mobile Device အသေးဆုံး Screen မှာ အနီရောင် ဘေးမျဉ်းလေးကိုမြင်ရမှာဖြစ်ပါတယ်။

For Tablet ဆိုတဲ့ **@media rules** ထဲကိုဝင်ဖတ်ကြည့်တဲ့အခါ စာသားအရောင်ကို စပြီး အနီရောင် code ကိုစခိုန်းထားပါတယ်။ ပြီးတော့ **text-align: center** ဆိုတဲ့အတွက် အလယ်ကို စရောက်ပါပြီ။ ဒီအထိကို tablet screen တွေက စလို့ စာသားက အလယ်ရောက်သွားပြီဆိုတာ စာဖတ်သူတို့ နားလည်လိမ့်မယ်လို့ ထင်ပါတယ်။

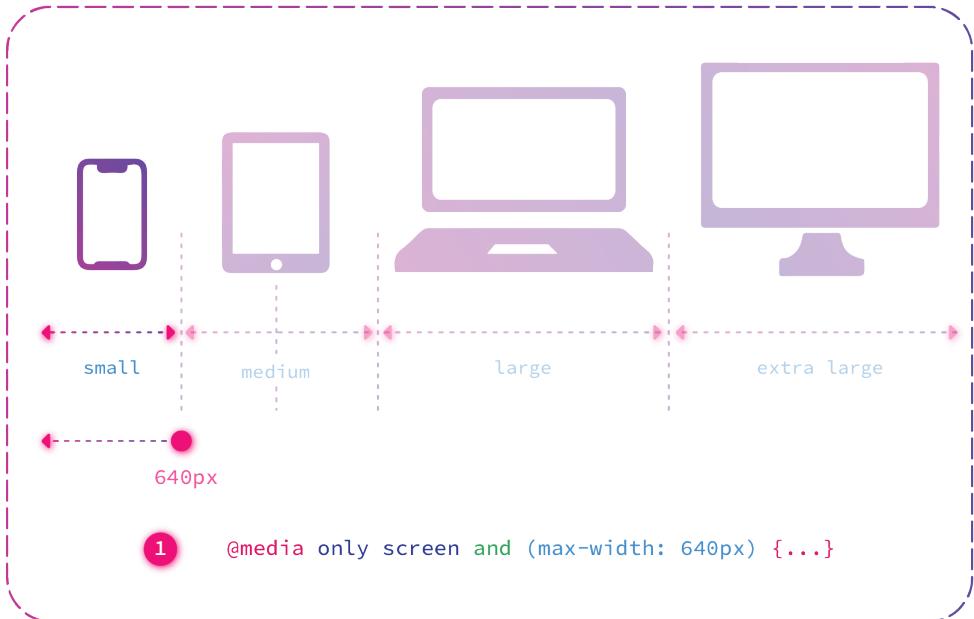
For Large Tablet ဆိုတဲ့ **@media rules** ထဲမှာတော့ **font-size: 46px** ဆိုပြီး ရေးထားတာကိုမြင်ရမှာပါ။ အဓိပ္ပာယ်ကတော့ Tablet ထဲ Screen Size ကြီးတယ့် Tablet တွေကစလို့ **Font Size 46px** နဲ့ပြုတော့မယ်လို့ ရေးသားထားခြင်းပဲဖြစ်ပါတယ်။

For Laptop and Upper Desktop ဆိုတဲ့ **@media rules** ထဲရောက်တဲ့အခါမှာတော့ **underlined** တားပေးလိုက်တဲ့ rule ကိုတွေ့မြင်ရပါပြီ။

.button class ကိုတော့ စာဖတ်သူတို့ကိုယ်တိုင် code ကိုဖတ်ပြီး နားလည်အောင် ကြိုးတားကြည့်ပါ...။

Responsive CSS Maximum Width ရေးသားနည်း

အခုရေးသားနည်းက `max-width` ရေးထုံးပဲဖြစ်ပါတယ်။ အများဆုံး ဒီ Screen အထိ ပဲဆိုတဲ့ အဓိပ္ပာယ်ပါ။ စာရေးသူအနေနဲ့ တော့ သိပ်သုံးလေ့မရှိပါဘူး။ အဓိကအားဖြင့် `min-width` နဲ့ အတူတူ တွဲပြီး `Between` အခြေအနေတွေမှာပဲ အသုံးများပါတယ်။



`(max-width: 640px)` အများဆုံး Mobile Screen 640px အထိပါပဲ။ ကျွန်ုန်းနောက်က Screen နေပြီးတော့ **CSS Rule** တွေသတ်ရောက်မှာမဟုတ်ပါဘူးလို့ သတ်မှတ်ထား ခြင်းဖြစ်ပါတယ်။ ရေးသားနည်းကို အောက်ပါ Code များမှာလေ့လာကြည့်ပါမယ်။

စာဖတ်သူတို့က `index.html` တွင် ဒီအောက်က Code ကိုကူးယူပြီးစမ်းကြည့်လို့ရပါမယ်။

<https://gist.github.com/lwinmoepaing/5a60182474db615b8251ae67670db48d>

`style.css` တွင် ဒီအောက်က Code ကိုကူးယူပြီးစမ်းကြည့်ပါ။

<https://gist.github.com/lwinmoepaing/ccfc8be9b05fe5be379c900c5c3bb7c5>

Codepen မှာလည်းတိုက်ရှိကဲ စမ်းကြည့်နိုင်ပါတယ်။

<https://codepen.io/lwinmoepaing/pen/yLZedYr>

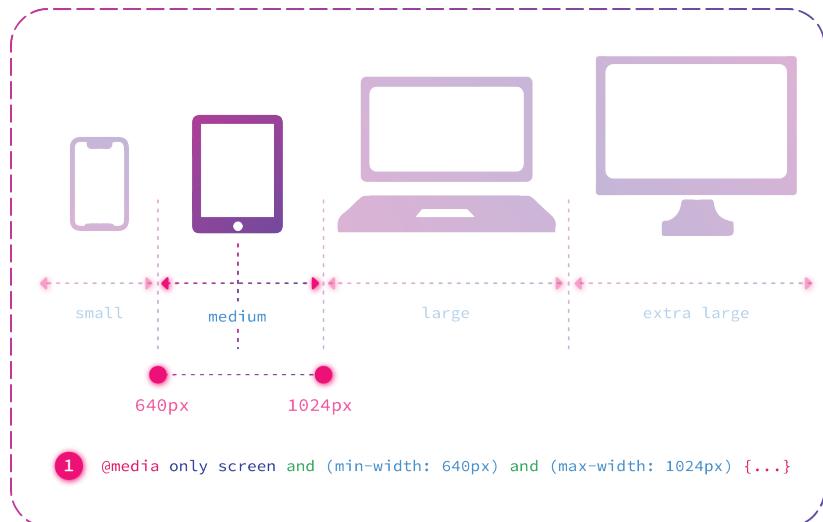
Browser ပေါ်တွင် Run လိုက်တဲ့အခါ Title Header က အလယ်မှာရှိနေတာကို တွေ့ရမှပါ။ Screen ကို ချုပြုကြည့်လိုက်ရင်တော့ နောက်ဆုံး ဘယ်ဘက် ဘေးအနီရောင်မျဉ်လိုင်း ဖြစ်သွားတာကို စာဖတ်သူတို့ကတွေ့ မြင်ရမှာဖြစ်ပါတယ်။

.title ကိုရေးထားတာကို For Only Phone ဆိုတဲ့ @media rules ထဲကိုဝင်ဖတ်ကြည့်တဲ့အခါ (max-width: 640px) ဖြစ်တဲ့အတွက် အထဲက Rules တွေက mobile screen အတွက်ပဲ ဖြစ်တယ်လို့ သတ်မှတ်ထားတာဖြစ်ပါတယ်။ စာသားအရောင်ကို ဘေးအနီရောင်မျဉ်လိုင်း နဲ့ စာသားဘယ်ဘက်ပဲထားထားတာဖြစ်ပါတယ်။

.title ကို Default အနေနဲ့က text-align: center ရယ် padding-left: 10px ကိုရေးထားတဲ့ အတွက် ကြောင့် mobile screen ကလွှဲရင်တော့ စာသားက အလယ်ရယ် padding-left ရယ်က သက်ရောက်သွားမှပါ။

Responsive CSS Between ထဲရေးသားနည်း

အခုရေးသားနည်းက For Tablets Only အတွက်လိုမျိုးမှာ ရေးသားခြင်းဖြစ်ပါတယ်။ ကြားထဲရောက်နေတဲ့ အခြေအနေမျိုးမှာပဲ ADS ပြတ္တမျို့၊ သို့မဟုတ် Designer တွေက အဲ ဘယ် Screen Size နဲ့ ဘယ် Screen Size ကြားထဲမှာပဲ တရာ့ရ ပြချင်တဲ့အခါမျိုးမှာ တွေ့ရများပါတယ်။ အောက်ကပုံလေးကို ကြည့်ရအောင် ...။



အနည်းဆုံး (`min-width: 640px`) ကနေ အများဆုံး (`max-width: 640px`) အတွင်း
သက်ရောက်မယ်လို့ ရေးထားတာဖြစ်ပါတယ်။

စာဖတ်သူတို့က `index.html` တွင် ဒီအောက်က Code ကိုကူးယူပြီးစမ်းကြည့်လို့ရပါမယ်။

<https://gist.github.com/lwinmoepaing/fa0ffd73463e640a8ca054318b14b975>

`style.css` တွင် ဒီအောက်က Code ကိုကူးယူပြီးစမ်းကြည့်ပါ။

<https://gist.github.com/lwinmoepaing/1cec1f65f467043560d3df6d6633be35>

Codepen မှာလည်းတိုက်ရိုက် စမ်းကြည့်နိုင်ပါတယ်။

<https://codepen.io/lwinmoepaing/pen/ZEwQgBe>

Browser ပေါ်တွင် Run လိုက်တဲ့အခါ **Title Header** က Tablet Screen ရောက်တဲ့အခါ
မှာသာ စာသားအနီး၊ အလယ်မှာရှိနေတာကို တွေ့ရမှာပါ။

Responsive Columns ရေးသားနည်း

Columns တွေကို စာဖတ်သူတို့က **Grid** အခန်းမှာလေ့လာပြီးဖြစ်ပါတယ်။ အခုခြုံအပိုင်း
မှာ ကျွန်ုပ်တော်တို့ရှိပြီးသား `column class` တွေကို **breakpoints** တွေခဲ့ပြီး စစ်ထုတ်
ရေးသားလိုက်မှာဖြစ်ပါတယ်။ အရင်ဆုံး Code တွေကို ကူးပြီး လေ့လာကြည့်ပေးပါ။

စာဖတ်သူတို့က `index.html` တွင် ဒီအောက်က Code ကိုကူးယူပြီးစမ်းကြည့်လို့ရပါမယ်။

<https://gist.github.com/lwinmoepaing/beb2ebfe3102b5b333115471af3fd7da>

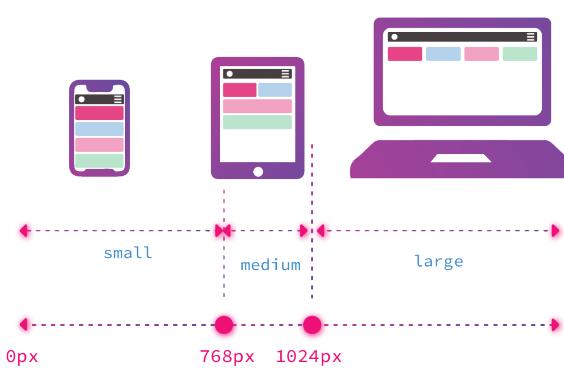
`style.css` တွင် ဒီအောက်က Code ကိုကူးယူပြီးစမ်းကြည့်ပါ။

<https://gist.github.com/lwinmoepaing/d0d846865b744c287668812f1c275cc1>

Codepen မှာလည်းတိုက်ရိုက် စမ်းကြည့်နိုင်ပါတယ်။

<https://codepen.io/lwinmoepaing/pen/NWoxQvG>

Browser ပေါ်တွင် Run လိုက်တဲ့အခါ အောက်ကပုံအတိုင်း Columns တွေက Screen အလိုက် ပြောင်းလဲဖော်ပြန်ပါမည်။



```
<div class="grid-12">
  <div class="box col-12 md:col-6 lg:col-3 min-h-60px">1</div>
  <div class="box col-12 md:col-6 lg:col-3 min-h-60px">2</div>
  <div class="box col-12 md:col-12 lg:col-3 min-h-60px">3</div>
  <div class="box col-12 md:col-12 lg:col-3 min-h-60px">4</div>
</div>
```

ပထမအနေနဲ့ `class` တစ်ခုကို လေ့လာပြီး ဘယ်လို အမိမျိုယ်နဲ့ သတ်မှတ်ထားလဲကြည့်ရအောင် ... ။ “`col-12 md:col-6 lg:col-3`”

Default အနေနဲ့ `col-12` ဆိုတော့ `Column 12` ခုလုံးယူပါမယ်။

`md:col-6` ဆိုတော့ `medium screen` တွေအတွက် `Column 6` နေရာယူသွားပါမယ်။

ဆိုလိုချင်တာက `Screen` ရဲ့ တစ်ဝါက်ကို ယူမယ်လို့ ဆိုလိုတာပါ။

`lg:col-3` ဆိုတာ `Large Screen` တွေအတွက် `Column 3` နေရာယူပါမယ်။

ဆိုပြီး သတ်မှတ်လိုက်တာပါ။ အခုလိုပြောတဲ့အတိုင်းပဲ ပထမဆုံး အနီးရောင် Box လေးက ယူသွားတာကိုလည်း ပုံတဲ့မှာတွေ့ရမှာပါ။

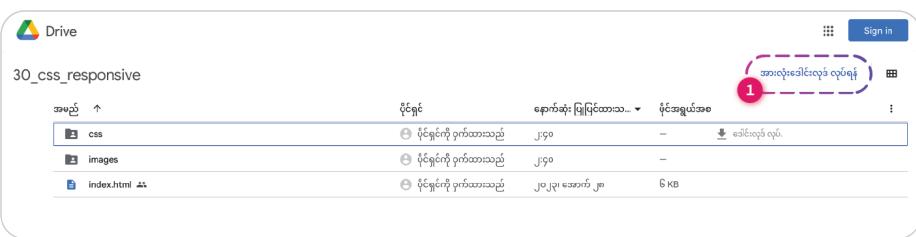
စာဖတ်သူတို့က `style.css` ထဲက `code` တွေကို လေ့လာကြည့်တဲ့အခါ ထူးခြားချက် အနေနဲ့ “`md\col-1`” ဆိုတဲ့ ရေးသားပုံလေးကိုမြင်ရမှာဖြစ်ပါတယ်။ “`\:`” လို့ဘာလို့ ရေးတာလည်းဆိုရင် `css` မှာ “`:hover`” တို့ဘာတို့နဲ့ နောက်မှာ `pseudo class` တွေက လိုက်ပါတယ်။ ကျွန်ုတ်တော်တို့က အဲပုံစံအတိုင်း `class` ကိုရေးချင်တဲ့အခါ “`:`” တိုက်ရိုက် ရေးလိုမရတဲ့အတွက် “`\:`” ကိုရေးကာခံးရေးရတာဖြစ်ပါတယ်။

Responsive LwinCafe

Grid အခန်းမှာ ပြုလုပ်ခဲ့တဲ့ LwinCafe ဆိုတဲ့ Website ကို Responsive အနေနဲ့ ရေးသားထားတဲ့ ကုတ်တွေကို စာဖတ်သူတို့ ဒေါင်းလော့ဆွဲပြီး ကိုယ်တိုင် Code ဖော်ပြီး လေ့လာကြည့်ပေးပါ။ တော်တော် နားလည်သွားပြီလို့ယူဆတဲ့အတွက် အဆင်ပြုမှာပါ။

ဒီအောက်က Drive Link မှာ အားလုံးဒေါင်းလုမ်း လုပ်ရန်ကိုနိုင်ပါ ပြီးရင် ကိုယ်ရဲ့ စက်ထဲ မှာ index.html ကိုဖွင့်လိုက်ပါ။

https://drive.google.com/drive/folders/15p0esAM30TUW9oMxntD_z9VI7dLo6B6e?usp=drive_link



Flex Layout ကိုလေ့လာခြင်း

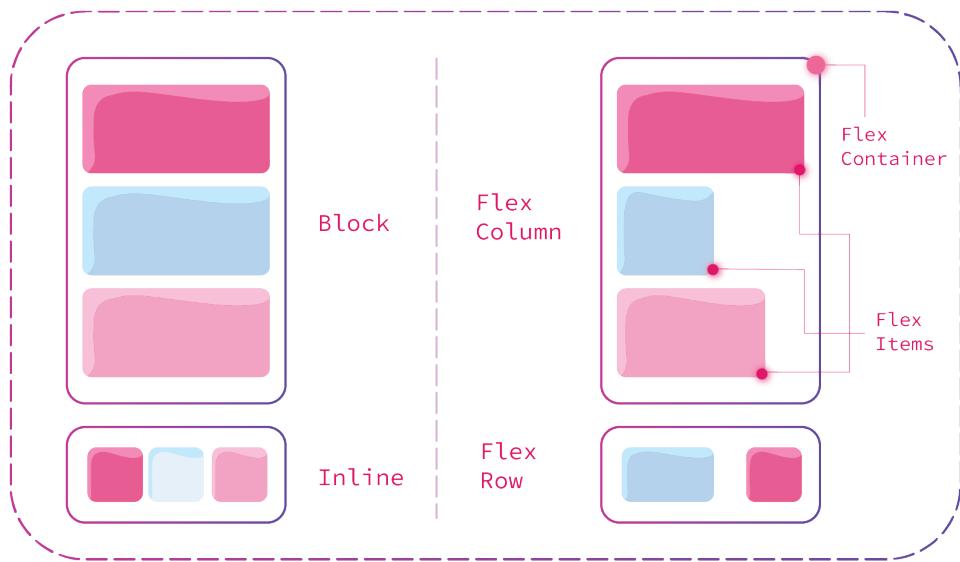
Flex ဘာလဲ । ဘာလို့သုံးလဲ ?

Powerful ဖြစ်တဲ့ Layout Model တစ်ခုဖြစ်ပါတယ်။ Grid Layout နဲ့မတူပဲ Flexible ဖြစ်တဲ့ Flex Layout ကို ဒီအပိုင်းမှာလေ့လာသွားကြမှာပါ။ Grid ကတော့ Web Page တစ်ခုလုံးကို Column တွေ၊ Row တွေနဲ့ Layout ချတဲ့နေရာမှာအသုံးများပေါ်ယဲ့၊ စာဖတ်သူတို့အတွက် အနုစိတ် လုပ်ရမယ့် Card တွေ Section Element တွေအတွက် Layout ကိုတော့ Flex Layout ဖြင့် အသုံးပြုရပိုလွယ်ပါတယ်။

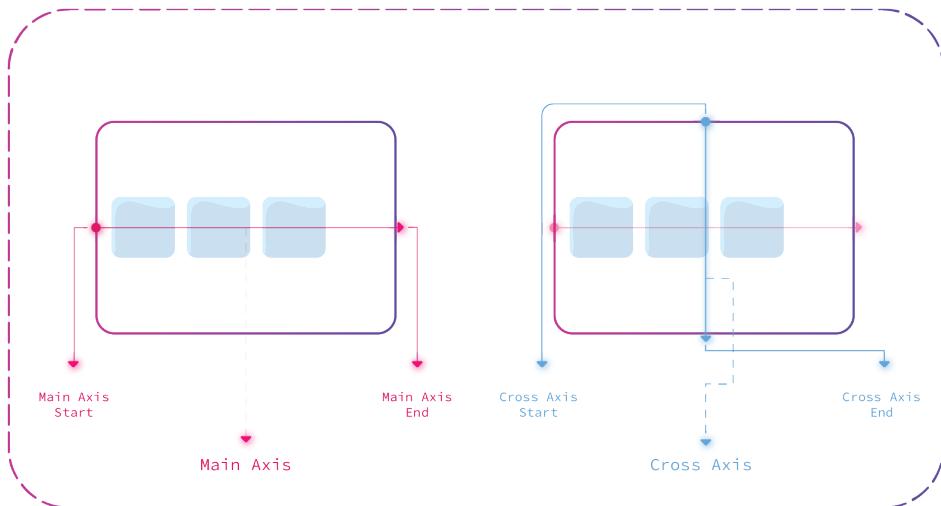
Flex က Complex ဖြစ်တဲ့ Layout တွေကို simplifies ဖြစ်စွာ အသုံးပြုနိုင်တာက တစ်ကြောင်း၊ mobile-friendly ဖြစ်နိုင်တာကြောင့်လည်း အသုံးများကြပါတယ်။ Flex မှာ အရေးကြီးတဲ့အချက်ကတော့ One Dimension ပုံစံနဲ့သွားပါတယ် သူမှာက Direction အနေနဲ့ Row (သို့) Column အနေနဲ့ တစ်ဖက်ပဲ ကျွန်ုတ်တို့က Direction သတ်မှတ် အသုံးပြုနိုင်ပါတယ်။ ဒါကို နောက်တစ်မျက်နှာမှာ လေ့လာသွားကြပါမယ်။

Flex မှာဘာတွေပါလဲ လေ့လာခြင်း

လေ့လာခြုံဖြိုးဖြစ်တဲ့ Block Element ၊ Inline Element တို့ကို မှတ်မိမိမယ်လို့ ယူစပါတယ်။ အောက်ကပုံလေးကို တစ်ခုကြိုးပေးပါ သာမန် Block Element တွေ ကတော့ နေရာအပြည့်ယူပြီး အောက်တစ်လိုင်းကပြန်စကြပါတယ်။ Inline Element တွေကတော့ ကိုယ်ရှိသလောက်နေရာပဲယူပြီး ညာဘက်ကပ်ရင် Horizontal အနေနဲ့ တည်ရှိသွားပါတယ်။ ညာဘက်ကပုံမှာတော့ Flex Layout ရဲ့ အကြမ်းဖျဉ်းပါပဲ။



Flex မှာ Parent က Flex Container ဖြစ်ပြီး၊ အတွင်းမှာ Flex Items ရယ်ဆိုပြီး ပါဝင်ပါတယ်။ Dimension အရ Column လို့သတ်မှတ်လိုက်ရင် အထက် Items တွေက ထောင်လိုက်မျဉ်းအတိုင်း အပေါ်ကနေ အောက်ကိုအစဉ်လိုက် ကျသွားမှာဖြစ်ပြီး၊ Row လို့သတ်မှတ်လိုက်ရင် အလျေားအလိုက်သတ်မှတ်ပေးသွားမှာပါ။ ဒီနေရာမှာ Inline Element တွေ ၊ Block တွေကို သုံးစရာမလိုတော့ဘူး။ Flex တစ်ခုတည်းနဲ့ ဘယ် Direction ကို Layout တွေ စီချင်သလဲဆိုတာကို သုံးနိုင်တာကိုမြင်စေချင်ပါတယ်။

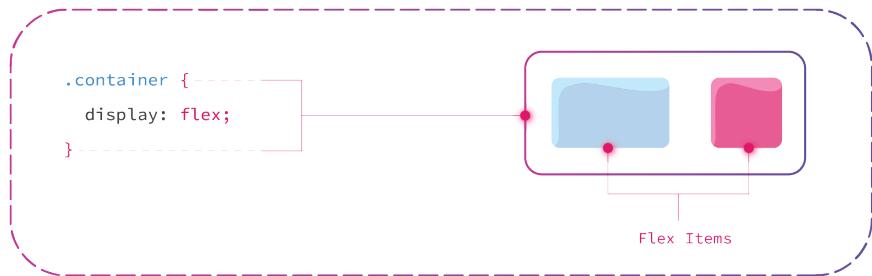


Flex Container ကို Row အနေနဲ့သတ်မှတ်လိုက်တဲ့အခါ ဘယ်ဘက်ပုံအရ အလျားလိုက် ဘယ်ဘက်ကနေ ညာဘက်သို့ Horizontal မျဉ်းတစ်ခုတည်ရှိသွားတာ ကို စာဖတ်သူတို့က မြင်ပေးရမှာဖြစ်ပါတယ်။ ထိုအရာကို MainAxis ဆိုပြီး ခေါ်ပါတယ် ပင်မဝန်ရှုံးပေါ့။ MainAxis ရဲ့ Direction အပေါ်မှာပဲ အထဲက Items လေးတွေက စီသွားမှာပါ။။ သူမှာ အစရယ် အဆုံးရယ်ရှိတဲ့အတွက် MainAxis Start နဲ့ MainAxis End ဆိုပြီးပါပါတယ်။။

ညာဘက်ပုံကတော့ CrossAxis ဆိုတာကို ဖော်ပြထားတာဖြစ်ပါတယ်။ CrossAxis က MainAxis ကို ဖြတ်လိုက်တဲ့ မျဉ်းကိုဆိုလိုတာပါ။။ သူမှာလဲ အစရယ် အဆုံးရယ်ရှိတဲ့အတွက် CrossAixs Start နဲ့ CrossAixs End ဆိုပြီးပါပါတယ်။။

စာဖတ်သူတို့က မမှတ်မိသေးလည်းပြသနာမရှိပါဘူး Flex လိုမြင်တာနဲ့ ဒီလို Axis တွေရှိတယ်ဆိုတာ နားလည်ထားရင်ရပါပြီ။။

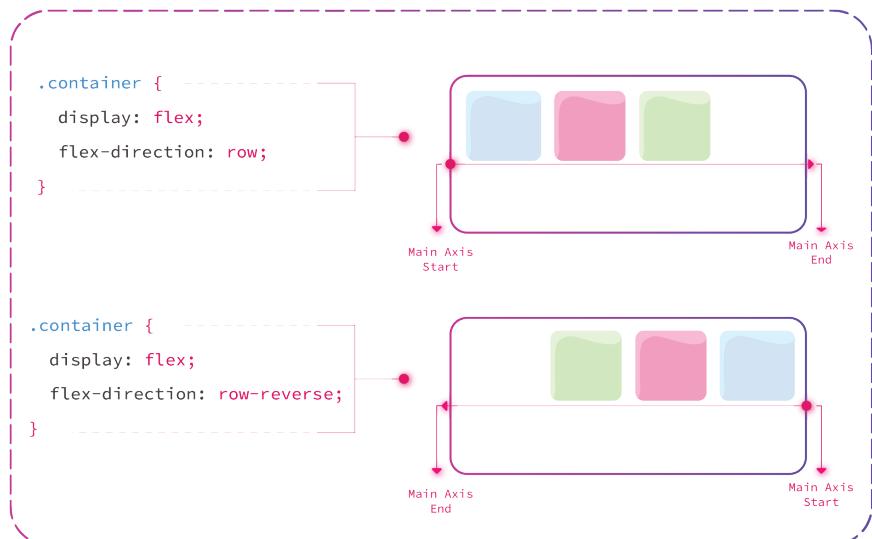
Flex Container အကြောင်း

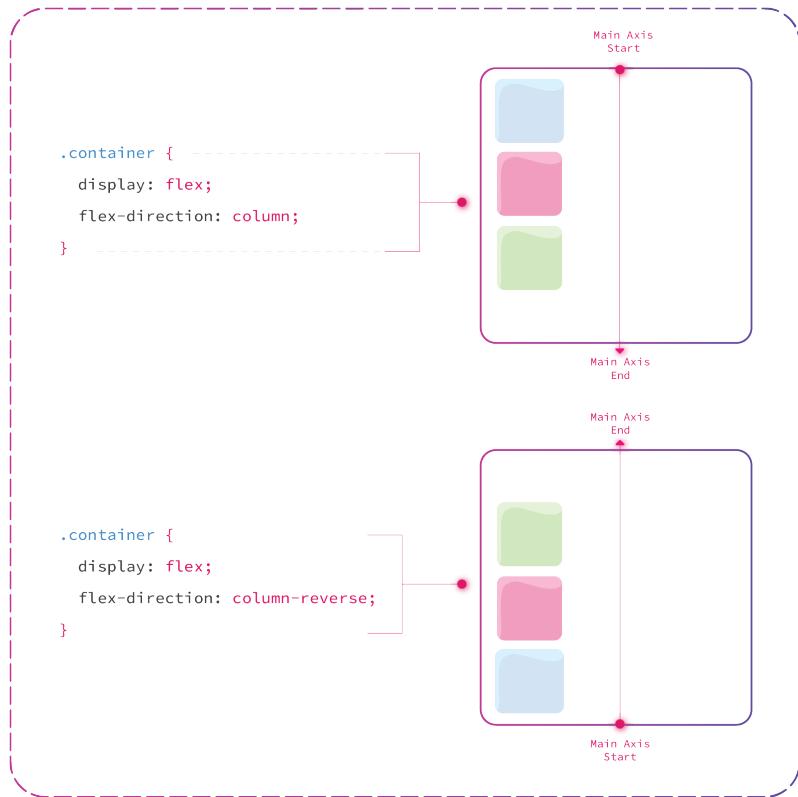


Flex Container ကို သတ်မှတ်စွဲအတွက် display: flex ဆိုပြီး သတ်မှတ်ပေးလိုက်ရင်ရပါပြီ။ အထူက Items တွေက အလိုလို Default အနေနဲ့ Horizontal လိုက်ဖြစ်သွားမှာပါ။

Flex Directions အကြောင်း

flex-direction ကို row ရယ် row-reverse ရယ်ကို အောက်ကပံအတိုင်း ထည့်ပြီး အသုံးပြန်ပါတယ်။ မြင်ရမှာတော့ Row ဆိုရင်တော့ MainAxis က Horizontal ဖြစ်နေပြီး Items တွေက အဲအပေါ်မှာ ဆီသွားမှာပါ။ အဲအတွက် ဘေးကပ်ရပ်ဆီသွားမှာ ဖြစ်ပါတယ်။ row-reverse ဆိုရင် MainAxis ရဲ့ အစကော်ဘက်ကနေ စသွားမှာပါ ဒုတိယပုံ မှာ မြင်ရတဲ့ အတိုင်းပြန် စီသွားမှာ ဖြစ်ပါတယ်။





flex-direction ကို column ရယ် column-reverse ရယ် အပေါ်ကပုံအတိုင်း
ထည့်ပြီးလည်း အသုံးပြန်စိတ်ယူမှတ်မှုများမှာဖြစ်ပြီး
Items တွေကထောက်လိုက် စီသွားမှာပါ။ column-reverse ဆိုရင် MainAxis မဲ့ အစက
အောက်ဘက်ကနေ စသွားမှာဖြစ်တဲ့အတွက် ပြောင်းပြန် Flex Items တွေက စီသွား
မှာဖြစ်ပါတယ်။ အောက်က Code တွေကို ကူးယူပြီးနောက် စာဖတ်သူတို့ စက်ပေါ်မှာ
လက်တွေ့ စမ်းသပ်ပြောင်းလဲ ရေးသားကြည့်ပေးပါ။

စာဖတ်သူတို့က `index.html` တွင် ဒီအောက်က Code ကိုကူးယူပြီးစမ်းကြည့်လို့ရပါမယ်။

<https://gist.github.com/lwinmoepaing/aaa16962133271ddecf04e13dfc9788a>

`style.css` တွင် ဒီအောက်က Code ကိုကူးယူပြီးစမ်းကြည့်ပါ။

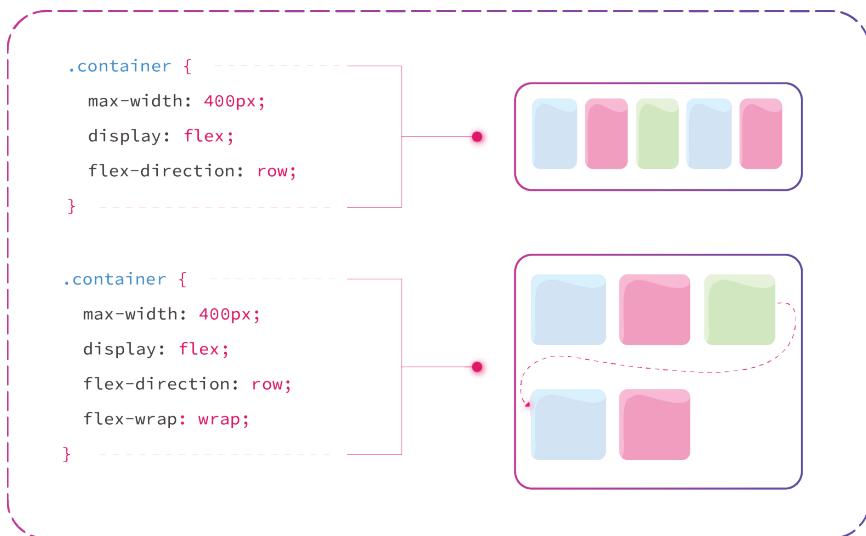
<https://gist.github.com/lwinmoepaing/4a83fdcc3c8f6eccfde97b922181c8ea>

Codepen မှာလည်းတိုက်ရိုက် စမ်းကြည့်နိုင်ပါတယ်။

<https://codepen.io/lwinmoepaing/pen/eYxZJby>

Flex Wrap အကြောင်း

Flex Wrap အကြောင်းမပြောခင် flex-direction ကို row ပေးလိုက်တဲ့ Example တစ်ခု အနေနဲ့ အထဲက Flex Items တွေက Horizontal အလိုက်စီသွားမှာဖြစ်ပါတယ်။ Parent ရဲ့ Width အကျယ် ကရိုသလောက် Items တွေက များလာတဲ့အခါ ပြုတဲ့သိပ်ပြီး ကပ် သပ်သွားမှာပါ။ ဒီအတွက် Flex က One Dimension လို့ပြောရတာလဲမမှားပါဘူး Row ဆိုရင် Row တရာ့တည်းပဲ Direction တစ်ခုတည်း ဦးတည်းလို့ပါ။ အောက်ကပိုလေး ကို လေ့လာကြည့်ပါမယ်။



Flex Container ရဲ့ Width အကျယ်ထက် ကျော်လွန်သွားတဲ့အခါ flex-wrap: wrap ကို အသုံးပြုပြီး အထဲက Items တွေကို ပုံပျက်ပျယ်မှုမရှိအောင် လုပ်နိုင်ပါတယ်။ အပေါ်မှာပြထားတဲ့ ပထမပုံထက် Items တွေက ပူးကျပ်နေမှုကနေ နှုတိယပုံကဲခဲ့ပါ။ Container Wrap လိုက်ပြီး အဆင်ပြေသွားအောင် ထိန်းညီလို့ရပါတယ်။

စာဖတ်သူတို့က [index.html](#) တွင် ဒီအောက်က Code ကိုကူးယူပြီးစမ်းကြည့်လို့ရပါမယ်။
<https://gist.github.com/lwinmoepaing/9523c29a19a0470e0dbd5d0de496a35a>

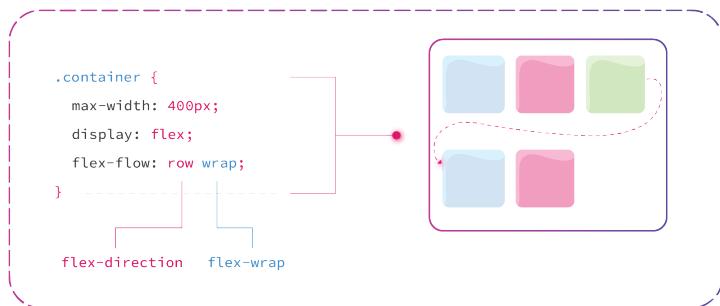
[style.css](#) တွင် ဒီအောက်က Code ကိုကူးယူပြီးစမ်းကြည့်ပါ။

<https://gist.github.com/lwinmoepaing/c928b8c405beafb159c71190c356b44d>

Codepen မှာလည်းတိုက်ရှိက စမ်းကြည့်နိုင်ပါတယ်။

<https://codepen.io/lwinmoepaing/pen/ZEwWWbx>

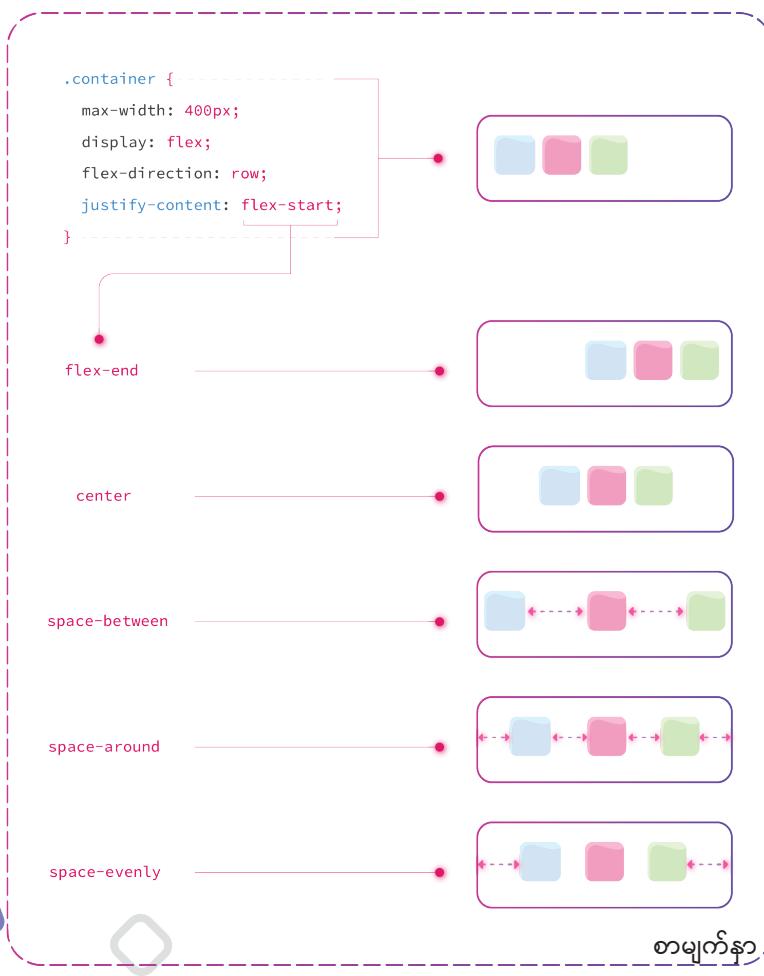
Flex Flow အကြောင်း



Flex Wrap နဲ့ Flex Direction အတွက် အတိုကောက် CSS Rules ကိုရေးတဲ့အခါ အသုံးပြုပါတယ်။

Justify Content အကြောင်းအရာ

Justify Content ကို မလေ့လာခင် အရင်ဆုံး အောက်ကုပ္ပါဒီ တစ်ချက်ကြည့်ပေးပါ ...။



Justify Content ဆိတာက Flex Container ကနေ အတွင်းက Flex Items တွေရဲ့ Layout ကို ဘယ်လိုအနေရာချမလဲနဲ့ လွှတ်နေစဲ Spacing တွေအတွက် သတ်မှတ်ပေးတဲ့ နေရာမှာအသုံးပြုပါတယ်။ အဓိကအားဖြင့်တော့ Main Axis အပေါ်မှာသာ Spacing ယူတာကို သတိထားရမှာပါ။

flex-start ဆိတာ Default Value ဖြစ်ပြီးတော့ MainAxis အစကနေစပြီ စီလိုက်တာပါ။

flex-end ဆိတာ MainAxis အဆုံးကနေစပြီ စီလိုက်တာပါ။

center ကတော့ items တွေကို center အလယ်ကျအောင် စီမံတာပဲဖြစ်ပါတယ်။

space-between အဓိပ္ပာယ်ကတော့ Item တွေကြားထဲမှာ spacing အပြည့် အကွာ အဝေးကို ထားမယ်လို့ သတ်မှတ်လိုက်တာပါ။ container နဲ့ items ကြားထဲမှာတော့ မရှိပါဘူး။

space-around အဓိပ္ပာယ်ကတော့ Item တွေကြားထဲမှာ တူညီတဲ့ spacing အကွာအဝေး ကို ထားမယ်လို့ သတ်မှတ်လိုက်တာပါ။ container နဲ့ items တွေကြားထဲမှာလဲ spacing ရှိနေပါမယ်။

space-evenly သူကတော့ container ဘက်နေ items တွေဘက်ကို ခြားတဲ့ Spacing နှစ်ခုကို ညီတူညီမှုအရင်ယူပါတယ်။ ပြီးတဲ့နောက်မှ items တွေကြားထဲ spacing ကို ထပ်ခဲ့တာပါ။

ဒါတွေဖတ်ပြီးသွားတဲ့အခါ ပုံကိုတစ်ချက်ပြန်လေ့လာကြည့်ပေးပါ ဒါဆိုရင် မျက်လုံးထဲ မြင်သွားပါလိမ့်မယ် ပြီးရင် စာဖတ်သူတို့က အောက်က Code တွေကို ကိုယ့်ရဲ့စက်မှာ Run ပြီး justify-content ရဲ့ တန်ဖိုးတွေကို ပြောင်းလဲအသုံးပြုပြီး လေ့ကျင့်ကြည့်ပါ။

စာဖတ်သူတို့က **index.html** တွင် ဒီအောက်က Code ကိုကူးယူပြီးစမ်းကြည့်လိုပါမယ်။

<https://gist.github.com/lwinmoepaing/774716927e58205fb085cc8d1c99c18>

style.css တွင် ဒီအောက်က Code ကိုကူးယူပြီးစမ်းကြည့်ပါ။

<https://gist.github.com/lwinmoepaing/1ef600d68979013690be00d58e55699b>

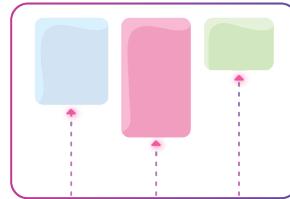
Codepen မှာလည်းတိုက်ရိုက် စမ်းကြည့်နိုင်ပါတယ်။

<https://codepen.io/lwinmoepaing/pen/rNPeLNE>

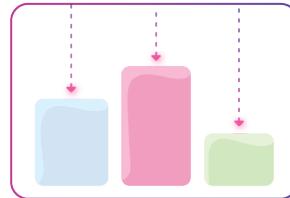
Align Items များအကြောင်းအရာ

Align Items သည်လည်းပဲ Flex Container ကနေ အတွင်းက Flex Items တွေရဲ့ Layout ကို နေရာချေပေးတဲ့အပါ မှာအသုံးပြုပါတယ်။ အခိုကအားဖြင့်တော့ Cross Axis အပေါ်မှာသာ Spacing ယူတာကို သတိထားရမှာပါ။

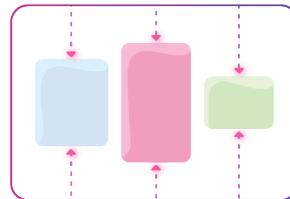
```
.container {
    max-width: 400px;
    display: flex;
    flex-direction: row;
    align-items: flex-start;
}
```



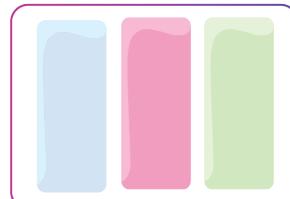
`flex-end`



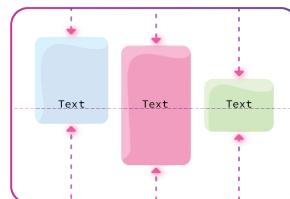
`center`



`stretch`



`baseline`



flex-start ဆိုတာ Default Value ဖြစ်ပြီးတော့ CrossAxis အစကနေစပြီ စီလိုက်တာပါ။

flex-end ဆိုတာ CrossAxis အဆုံးကနေစပြီ စီလိုက်တာပါ။

center ကတော့ items တွေကို center အလယ်ကျအောင် စီမံတာပြုခြင်ပါတယ်။

stretch အဓိပ္ပာယ်ကတော့ Item တွေကြားထဲမှာ spacing လုံးဝမထားတော့ပဲ အပြည့်လှုပျင်တဲ့အခါမှာ အသုံးများပါတယ်။

baseline အဓိပ္ပာယ်ကတော့ Text Baseline ကို Align ထိုပြီး spacing တွေချိန်ညွှတ်တာပဲ ဖြစ်ပါတယ်။ Text Baseline ဆိုတာထက် content baseline ဆိုပိုမျိန်ပါတယ် အလွယ် အနေနဲ့တော့ Text အနေနဲ့ပဲ မြင်ထားပေးရင်ပဲရပါပြီ။

စာဖတ်သူတို့က [index.html](#) တွင် ဒီအောက်က Code ကိုကူးယူပြီးစမ်းကြည့်လို့ရပါမယ်။

<https://gist.github.com/lwinmoepaing/68ea0282146547b5039cf2b5ef25fe67>

style.css တွင် ဒီအောက်က Code ကိုကူးယူပြီးစမ်းကြည့်ပါ။

<https://gist.github.com/lwinmoepaing/6ff4a47db49690213605b4eb6d44f497>

Codepen မှာလည်းတိုက်ရှိက် စမ်းကြည့်နိုင်ပါတယ်။

<https://codepen.io/lwinmoepaing/pen/wvNGGjq>

Flex Gap များအကြောင်းအရာ

Gap တွေက Grids အခန်းမှာလည်း စာဖတ်သူတို့လေ့လာပြီးဖြစ်ပါတယ်။ row တို့ column တို့ကြားထဲက အကွာအဝေးကို ပေးချင်တဲ့အခါမျိုးရှာသုံးတာပဲ ဖြစ်ပါတယ်။ အခုလည်း Flex Items တွေရဲ့ ကြားအကွာအဝေးကို သတ်မှတ်ချင်တဲ့အခါ အသုံးပြုပဲ ကိုလေ့လာသွားမှာဖြစ်ပါတယ်။

နောက်တစ်မျက်နှာမှာ ကျွန်ုတ်တို့ ဆက်ပြီးလေ့လာကြည့်ရအောင်။

```
.container {
  max-width: 400px;
  display: flex;
  flex-flow: row wrap;
  row-gap: 20px; ----->
  column-gap: 20px; ----->
}

gap: 20px 20px;
```

row-gap ကိုအသုံးပြုပြီး Rows တွေရဲ့ ကြားတဲ့ spacing ကို သတ်မှတ်နိုင်ပြီး၊ Column အတွက် column-gap ကိုအသုံးပြနိုင်ပါတယ်။ အတိုကောက်အနေနဲ့ gap ဆိုပြီး တစ်ခုတည်းကို သုံးနိုင်ပါတယ်။ gap: 20px 20px; ဆိုရင် အရေ့က 20px က row-gap ကို ကိုယ်တူးပြုပြီး၊ ဒုတိယ 20px ကတော့ column-gap ကိုကိုယ်တူးပြုတော့ဖြစ်ပါတယ်။

တဖတ်သူတို့က [index.html](#) တွင် ဒီအောက်က Code ကိုကူးယူပြီးစမ်းကြည့်လို့ရပါမယ်။
<https://gist.github.com/lwinmoepaing/ff00d47d1648d03a12912a26ae0aeab1>

[style.css](#) တွင် ဒီအောက်က Code ကိုကူးယူပြီးစမ်းကြည့်ပါ။

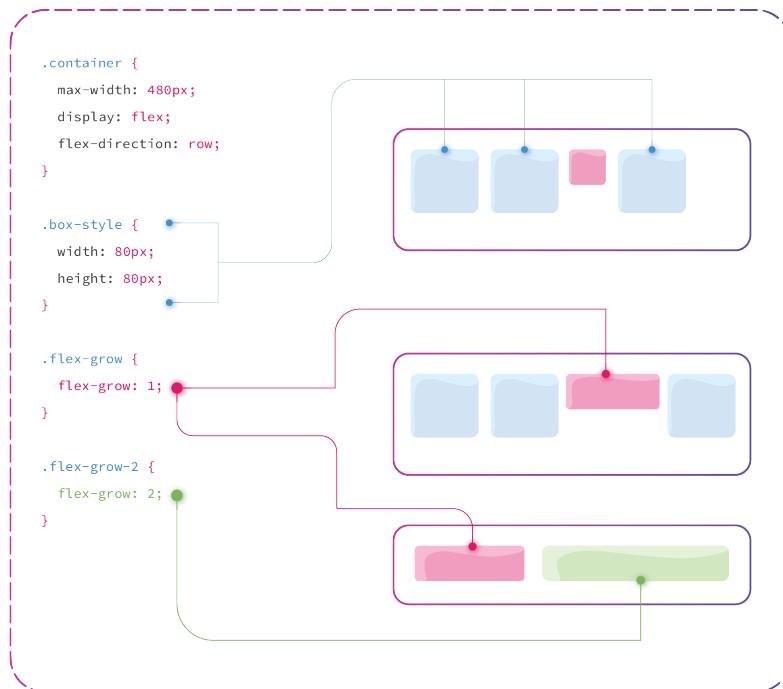
<https://gist.github.com/lwinmoepaing/9ba10a4dd88671f143ddca0c8f92e65f>

Codepen မှာလည်းတို့ကိုရှိပါ စမ်းကြည့်နိုင်ပါတယ်။

<https://codepen.io/lwinmoepaing/pen/VwgajeP>

Flex Grow များအကြောင်းအရာ

Flex Grow က Container Level ကနေမဟုတ်ဘဲ ကိုယ့်ရဲ့ Flex Items အတွင်းကနေပြီးတော့ နေရာ spacing ကိုတွေယူမယ့်အခါမှာသုံးတာပါ။ ဒဲတော့ class တွေကို item တွေကနေ သတ်မှတ်ပြီး ကိုယ်ဘယ်လောက် နေရာတွေယူသွားမလဲဆိုပြီး အသုံးပြုလေ့ရှိပါတယ်။



ပုံထဲက အတိုင်း box-style: 80px ရှိတဲ့ အားလုံးရောင် Box 3 ခုနဲ့ Content အတိုင်းပဲရှိတဲ့ content ရှိရင် ရှိသလောက်ပဲပြုမယ့် ပန်းရောင် Box လေးတစ်ခုရှိပါမယ်။ Content အတွက်မှာ နည်းနေတဲ့အတွက် Box က သေးသေးလေးအနေနဲ့ပဲပြုပါမယ်။ အဲဒီ Item ထဲကို flex-grow:1 ဆိုပြီးထည့်လိုက်တဲ့အခါမှာတော့ Container ထဲမှာ လွတ်နေတဲ့ space တွေကို သူက ရရှိသွားမှာဖြစ်ပါတယ်။ အဲလို့ အနေအထားမျိုးမှာသုံးကြလေးရှိတယ်။

ဒုတိယပုံအရ flex-grow-1 ဆိုတဲ့ class နဲ့ flex-grow-2 ဆိုတဲ့ Box (2) ခုကိုပဲ အသုံးပြုထားတာရို့ လေ့လာကြည့်ရအောင် ...။ width တွေ height တွေ သတ်မှတ်မထားဘူးဆိုရင် အထဲက Items တွေကို flex-grow (1 နဲ့ 2) နဲ့က ကျွန်တော်တို့ Grid မှာလေ့လာဖူးတဲ့ fr unit နဲ့ ဆင်တူပါတယ်။ 1 + 2 ပေါင်းလဒ်က 3 ဖြစ်ပါတယ်။ တစ်ခါက (2) ပုံပုံရင် တစ်ပုံ ယူမှာဖြစ်ပြီး flex-grow: 2 က တော့ (၂) ပုံကို ယူသွားမှာပါ။ ဒီလောက်ဆိုနည်းနည်းသဘောက်ပေါက်သွားမယ်လို့ ယုံကြည့်ပါတယ်။

စာဖတ်သုတေသန index.html တွင် ဒီအောက်က Code ကိုကူးယူပြီးစမ်းကြည့်လို့ရပါမယ်။
<https://gist.github.com/lwinmoepaing/f3977898ecde745cff40a4cda974f87>

style.css တွင် ဒီအောက်က Code ကိုကူးယူပြီးစမ်းကြည့်ပါ။
<https://gist.github.com/lwinmoepaing/81fee140e96711029575c398a0e12ea4>

Codepen မှာလည်းတိုက်ရှိက် စမ်းကြည့်နိုင်ပါတယ်။

<https://codepen.io/lwinmoepaing/pen/eYxZzym>

Flex Align Self အကြောင်းအရာကိုလွှဲလာခြင်း

```
.container {
    max-width: 480px;
    display: flex;
    flex-direction: row;
}
```

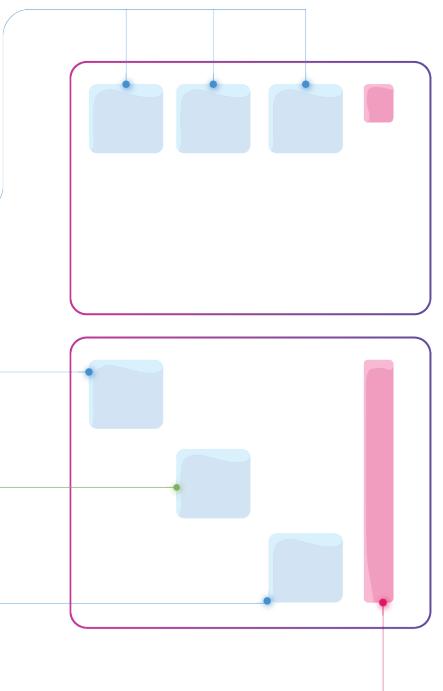
```
.box-style {
    width: 80px;
    height: 80px;
}
```

```
.align-self-start {
    align-self: flex-start;
}
```

```
.align-self-center {
    align-self: center;
}
```

```
.align-self-end {
    align-self: flex-end;
}
```

```
.align-self-stretch {
    align-self: stretch;
}
```



Align Items self သည် Flex Items တွေ၏ Layout ကို နေရာချပေးတဲ့အခါ မှာ အသုံးပြုပါတယ်။ အဓိကအားဖြင့်တော့ Cross Axis အပေါ်မှာသာ Spacing ယူတာ ကို သတိထားရမှာပါ။ အဲတော့ class တွေကို item တွေကနေ သတ်မှတ်ပြီး ကိုယ် ဘယ်လောက် နေရာတွေယူသွားမလဲဆိုပြီး အသုံးပြုလေ့ရှိပါတယ်။ Flex Align Item နဲ့ အတူတူပဲ ဖြစ်တာကြောင့် ဒါကိုတော့မရင်းပြတော့ပါဘူး အတူတူပါပဲ Item ကနေ သတ်မှတ်တာပဲကွာသွားတာဖြစ်ပါတယ်။

ကျွန်ုတ်တို့ စာအုပ်ကို ဒီနေရာမှာပဲ အဆုံးသတ်လိုက်ပါတော့မယ်။ အားလုံး အဆင်ပြေကြပါစေခင်ဗျာ ...။

နိဂုံးချုပ်

ဒီစာအပ်က Web Development ကိုအခုံမှစလေ့လာမယ့်သူတွေအတွက် Visual အသားပေးပုံစံလေးနဲ့ ကျွန်ုင်တော် ရေးသားဖြစ်ခဲ့တာပါ။ ရည်ရွယ်ချက်ကတော့ Absolute Beginner တွေ စတင်လျောက်မယ့် လမ်းမှာ အကျိုးတစ်ခုခုရရှိသွားရင်ပဲ ကျေန်ပဲပါတယ်။ ဒါကြောင့် နောက်ဆက်တွဲလေ့လာရမယ့် JavaScript နည်းပညာ တွေကို ဆက်လက် လေ့လာသွားကြဖို့ တိုက်တွန်းချင်ပါတယ်။

ဒီစာအပ်ရဲ့ အကူအညီနဲ့ ဆက်လက်ပြီး ကိုယ့်ရဲ့ Roadmap ကိုအကောင်အထည်ဖော် နိုင်ဖို့ ဆုမွန်ကောင်းတောင်းပေးလိုက်ရပါတယ်ခင်ဗျာ။

ကျေးဇူးပါယျ။

၂၀၂၃ ခုနှစ်၊ အောက်တိုဘာ (၃၀) ရက်နေ့တွင် ရေးသားပြီးစီးပါသည်။

ဆက်သွယ်ရန် -

mail: lwinmoepaing.dev@gmail.com

ph: 09420059241

social: <https://fb.com/lwin.im>