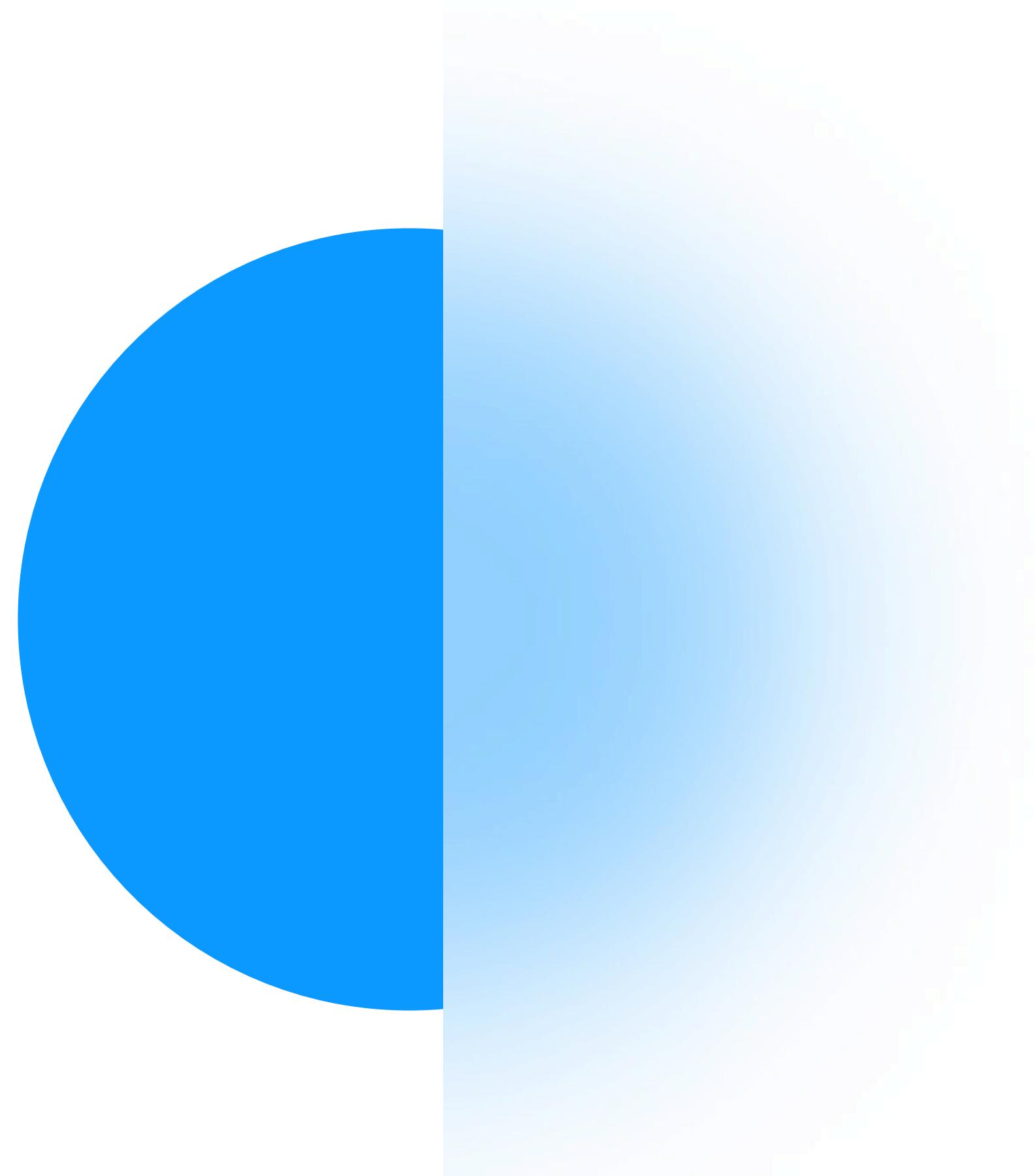


CSS (Cascading Style Sheets)

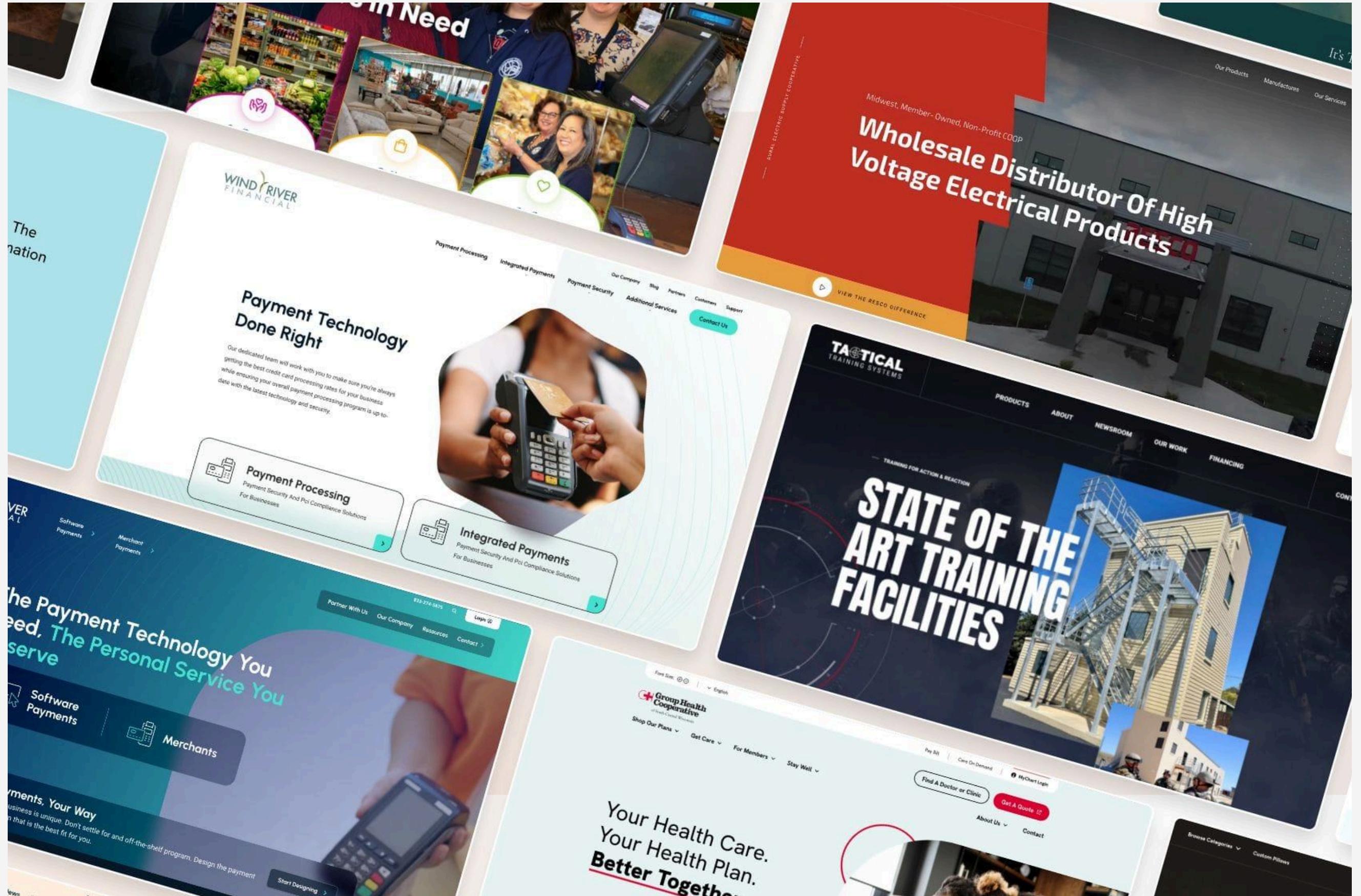
Les 2 Typography & Colors

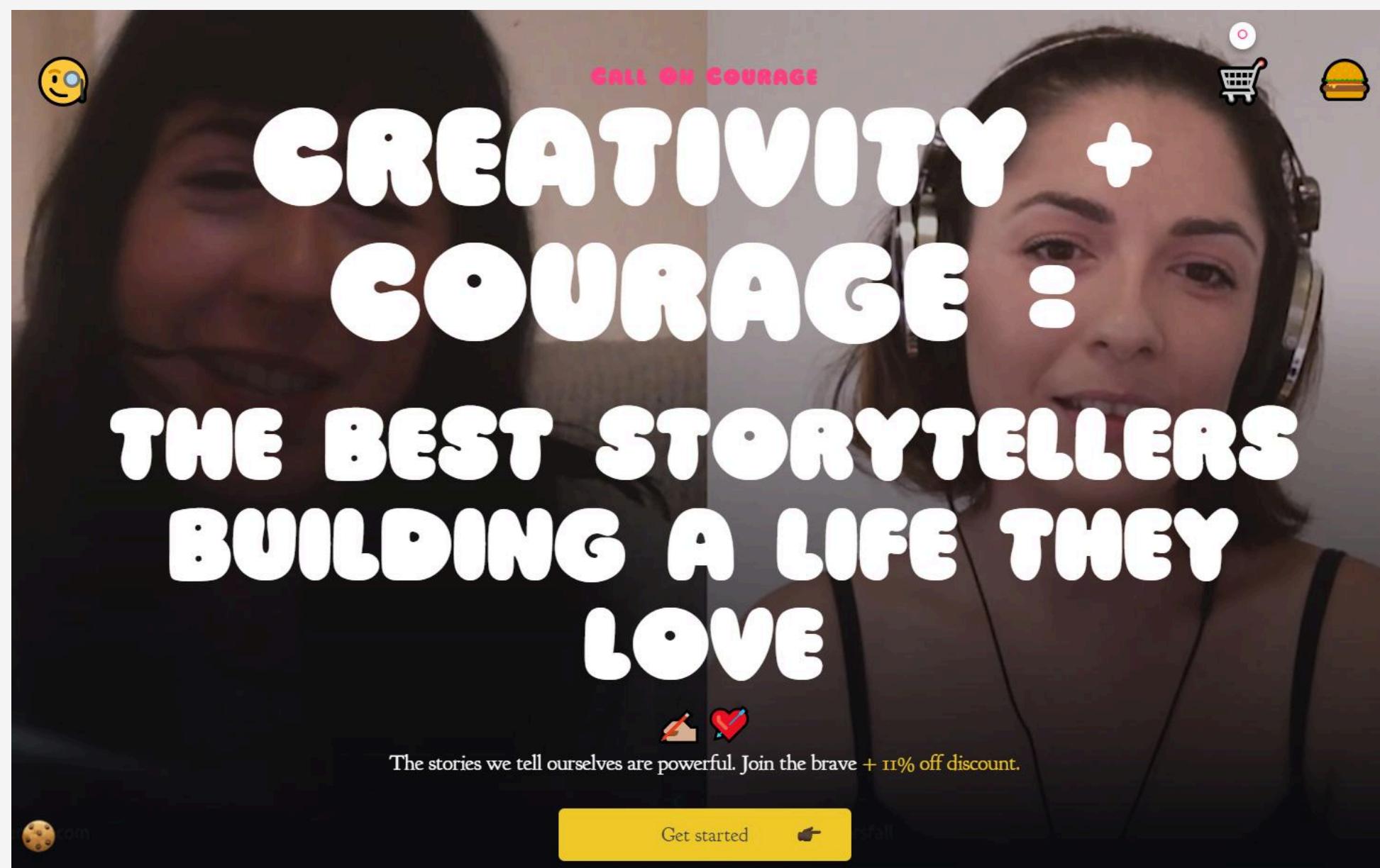


الموضوع	الوقت (التوقيت السوري)	التاريخ
HTML: Basics & Elements & page anatomy	12:00	2025\3\1
HTML: Basics check	12:00	2025\3\8
CSS: Basics & Selectors	12:00	2025\3\15
Typography & Colors	12:00	2025\3\22
Positions & :has() :not()	12:00	2025\3\29
Flexbox	12:00	2025\4\5
Grid	19:30	2025\4\7
States & Media Querie	19:30	2025\4\9
Animations	19:30	2025\4\11
JS: Basics & Events & First function (html+css+js)	19:30	2025\4\14
JS: Conditional Statements & Arrays & Loops	19:30	2025\4\16
JS: API & Fetch data (pokeapi)	19:30	2025\4\18
Last check	19:30	2025\4\20

مقدمة

للخطوط و الالوان تأثير على تجربة المستخدم و
الشعور العام للتصميم
حيث تساعد الخطوط و الالوان الى جذب انتباه
المستخدم الى اجزاء معينة ذات اهمية او إيصال
رسالة|فكرة معينة





الخطوط

مثال:

الخطوط تساعده على:

- توجيه انتباه القراء إلى الكلمات الرئيسية والأزرار الموجودة
- إيصال شخصية العلامة التجارية (جاد | مرح ... الخ)

حيث يتم الأخذ بعين الاعتبار عند اختيار الخط:

- غرض الموقع وهدفه
- الجمهور المستخدم
- هوية العلامة التجارية
- الوضوح وقابلية القراءة (قياس الخط، ثخانة الخط، المسافات بين الأحرف و الكلمات و السطور ... الخ)



```
1 color:;  
2 font-family:;  
3 font-size:;  
4 font-variant:;  
5 font-weight:;  
6 font-style:;  
7  
8 text-align:;  
9 text-decoration:;  
10 text-transform:;  
11 text-indent:;  
12 text-shadow:;  
13  
14 line-height:;  
15 letter-spacing:;  
16 word-spacing:;  
17 writing-mode:;
```

Typography

خصائص CSS الرئيسية للتحكم في النصوص:

- تحديد الخطوط المفضلة: font-family:
- تغيير حجم الخط باستخدام وحدات مثل px, em, rem, %
- التحكم في سماكة الخط: font-weight: (normal, bold, 100-900)
- تحويل النص إلى عادي (normal) أو مائل (italic)
- تحديد المسافة بين الأسطر لتحسين قابلية القراءة: line-height:
- تعديل المسافات بين الحروف والكلمات: letter-spacing و word-spacing:
- محاذاة النص: text-align: (left, center, right, justify)
- إضافة تأثيرات مثل التسطير او ازالتها: text-decoration:

طرق تضمين الخطوط:

web safe fonts .1

font services .2

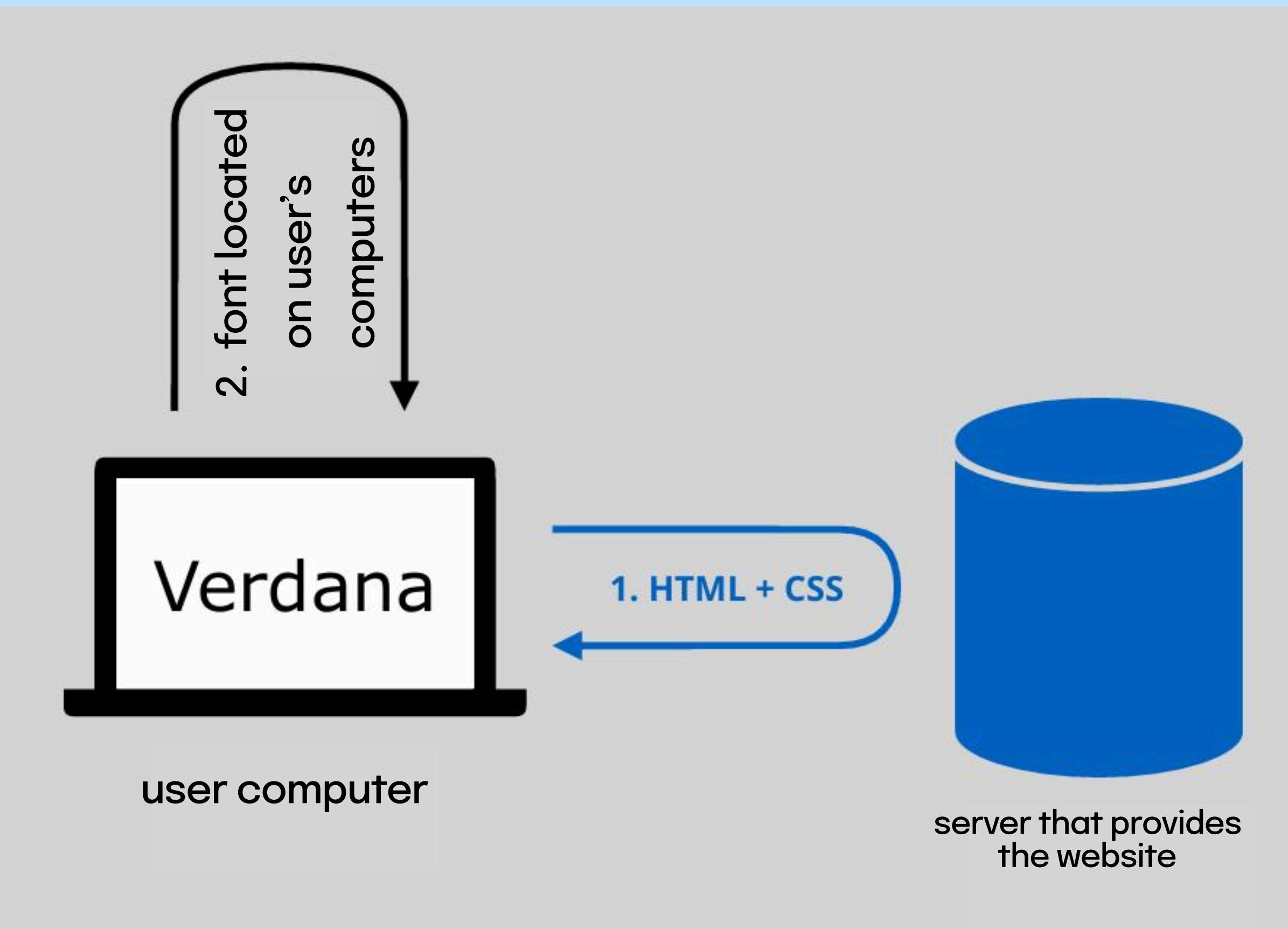
@font-face .3

طرق تضمين الخطوط:

web safe fonts .1

خطوط الويب الآمنة عبارة عن مجموعة من الخطوط التي يتم تثبيتها افتراضياً على جميع أنظمة التشغيل تقريباً. إذا كنت تستخدمها، فيمكنك على الأرجح افتراض أن الخط سيتم عرضه بشكل صحيح. لكي تكون في الجانب الآمن، أضف دائمًا إحدى عائلات الخطوط العامة كخط احتياطي

(Serif، أو Sans-serif، أو Monospace، أو Cursive، أو Fantasy).



HTML

```
<h1>Een font</h1>
```

CSS

```
h1 {  
    font-family:Verdana, Arial, sans-serif;  
}
```

Use Verdana if it's available

Use Arial if Verdana is not available

Use sans-serif if Verdana and Arial are not available

طرق تضمين الخطوط:

web safe fonts .1

ملخص web safe fonts

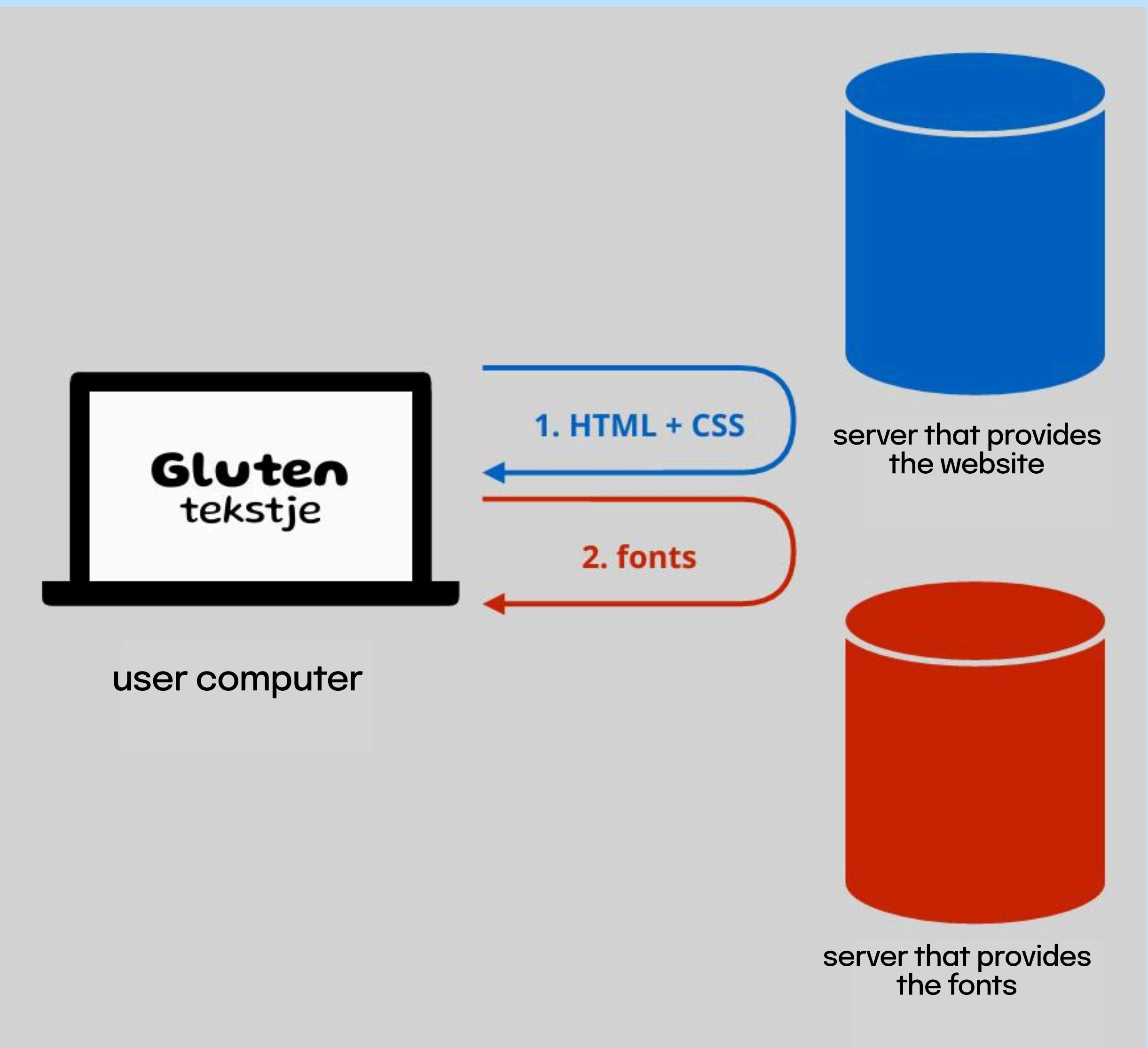
<https://web.mit.edu/jmorzins/www/fonts.html>

طرق تضمين الخطوط:

font services .2

ملخص web safe fonts

<https://web.mit.edu/jmorzins/www/fonts.html>



HTML

```

<head>
  <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=Gluten&display=swap" rel="stylesheet">

  <link href="/styles/style.css" rel="stylesheet">
</head>

<body>
  <h1>Gluten</h1>
  <p>tekstje</p>
</body>

```

CSS

```

h1, p {
  font-family: 'Gluten', cursive;
}

```

Code that you also get from Google Fonts

The stylesheet where you use the fonts

طرق تضمين الخطوط:

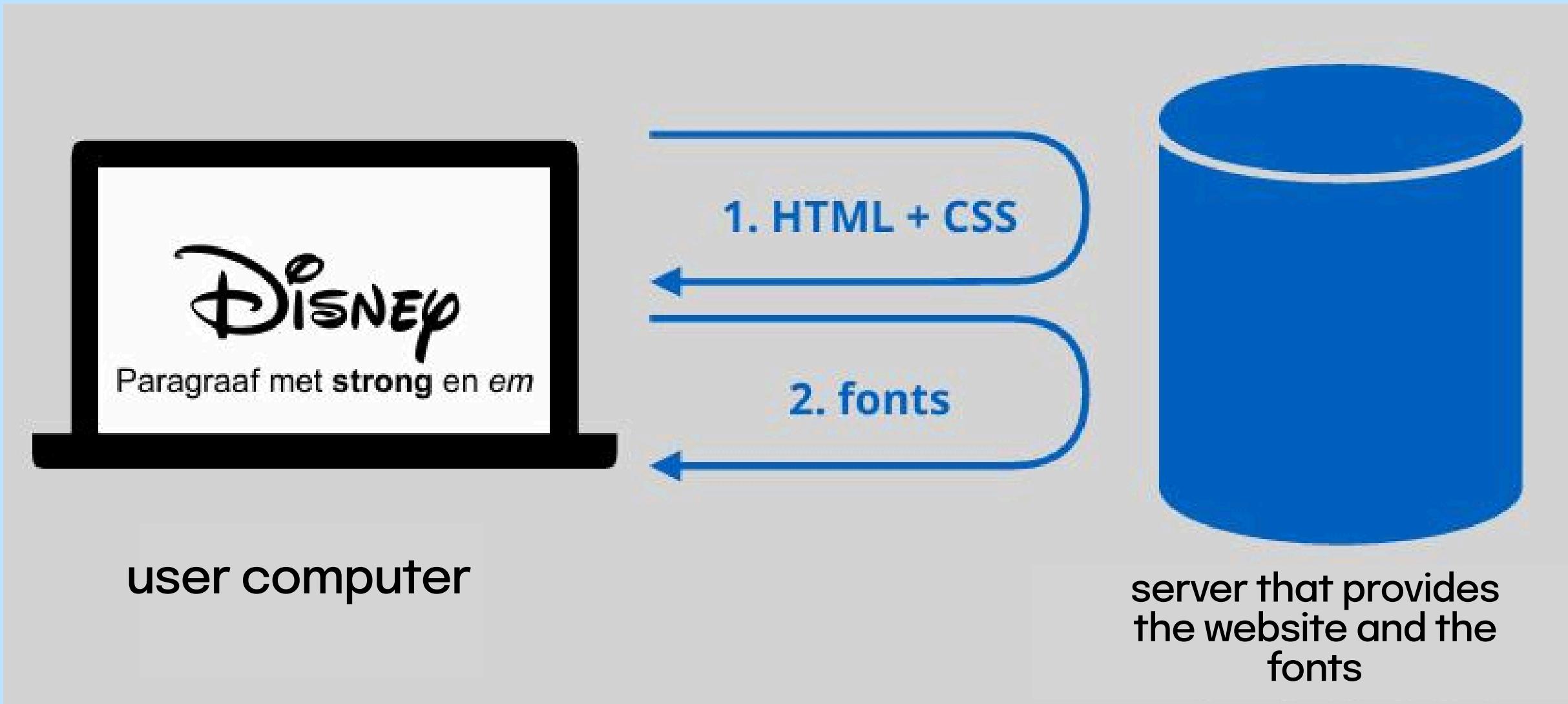
font services .2

يمكنك تنزيل الخطوط من خدمات الخطوط عن طريق إضافة جزء صغير من الكود إلى HTML (أو CSS أو JS). وبعد ذلك يمكنك استخدامها في CSS الخاص بك.

طرق تضمين الخطوط:

@font-face .3

1. ضع الخطوط في مجلد الخطوط الخاص بك.
2. قم بتضمين كود @font-face في CSS الخاص بك حتى يتم تنزيل الخطوط على جهاز الكمبيوتر الخاص بالمستخدم.
3. ثم يمكنك استخدام الخطوط





```

1  @font-face {
2      /* الاسم اختياري لكن من الأفضل التسمية باسم الخط ذاته */
3      font-family: "Roboto Mono";
4      /* تحديد نوع الخط ومكان وجوده في ملفات الموقع */
5      src:
6          url(fonts/roboto-mono-v12-latin-regular.woff2) format("woff2"),
7          url(fonts/roboto-mono-v12-latin-regular.woff) format("woff"),
8          url(fonts/roboto-mono-v12-latin-regular.ttf) format("truetype");
9  }
10
11 @font-face {
12     font-family: "Roboto Mono";
13     src:
14         url(fonts/roboto-mono-v12-latin-700italic.woff2) format("woff2"),
15         url(fonts/roboto-mono-v12-latin-700italic.woff) format("woff"),
16         url(fonts/roboto-mono-v12-latin-700italic.ttf) format("truetype");
17     /* تحديد وزن الخط */
18     /* تحديد نوع الخط */
19 }
20
21 h1,
22 p {
23     font-family: "Roboto Mono", monospace;
24 }
25
26 h1 {
27     font-family: "Roboto Mono", monospace;
28     font-weight: 700;
29     font-style: italic;
30 }
31

```

طرق تضمين الخطوط:

@font-face .3

1. ضع الخطوط في مجلد الخطوط الخاص بك.
2. قم بتضمين كود font-face@ CSS الخاص بك حتى يتم تنزيل الخطوط على جهاز الكمبيوتر الخاص بالمستخدم.
3. ثم يمكنك استخدام الخطوط

تطلب أنظمة التشغيل المختلفة أنواعاً مختلفة من الخطوط. في السابق كانت هذه، على سبيل المثال، (eot و ttf و woff). في الوقت الحاضر، يتم تفضيل woff2 و woff.

تتيح لك أداة [font-squirrel](#) تحويل الخط الخاص بك إلى أنواع الخطوط المختلفة التي تحتاجها. أثناء العملية، سيتم أيضًا إنشاء CSS الذي تحتاجه.

font-display:

- يستخدم هذا السلوك الافتراضي للمتصفح، والذي سيختلف: **auto**.
- يتم إخفاء النص أولاً لفترة قصيرة، ولكنه سيتغير إلى الخط المخصص عندما يصبح متاحاً. يُقال إن هذه القيمة لها فترة تبديل غير محدودة
- لا يتم إخفاء النص أبداً ويتغير إلى الخط المخصص عندما يصبح متاحاً، يوفر هذا أيضاً فترة تبديل غير محدودة
- **fallback**: يتم إخفاء النص لفترة قصيرة جداً (فترة الحظر)، ثم تكون هناك فترة تبديل قصيرة. إذا لم يتم تحميل الخط المخصص خلال فترة التبديل، فلن يتم تحميله على الإطلاق
- **optional**: يتم منح النص فترة حظر قصيرة جداً للتحميل (~ 100 ملي ثانية). إذا لم يتم تحميل الخط خلال فترة الحظر هذه، فسيتم استخدام الخط البديل ولن يتم تحميل الخط المخصص على الإطلاق. ومع ذلك، لا يزال الخط يتم تنزيله وتخزينه مؤقتاً خلف الكواليس. وهذا يعني أنه في عمليات تحميل الصفحات اللاحقة، سيصبح الخط المخصص متاحاً في ذاكرة التخزين المؤقت ثم يتم تحميله على الفور

طرق تضمين الخطوط:

3. استخدام font-display للتحكم في تحميل الخط @font-face

باستخدام "font-display"، يمكننا التحكم في كيفية تحميل الخطوط بدقة. ويمكن أن يؤدي هذا إلى تحسين تجربة المستخدم بشكل كبير عند استخدام الخطوط المخصصة.

في بعض الأحيان، عند استخدام الخطوط المخصصة، قد يواجه المستخدم إما FOUT (ظهور نص غير منسق) أو FOIT (ظهور نص غير مرئي) عند تحميل الصفحة لأول مرة. تختار بعض المتصفحات عرض النص على الفور، حتى إذا لم يتم تحميل الخط المخصص. سيعود المتصفح إلى الخط المخصص بمجرد تحميله بالكامل، ولكن هذا يؤدي إلى إنشاء FOUT. ستختفي متصفحات أخرى النص لفترة قصيرة حتى يتم تحميل الخط المخصص، مما يتسبب في حدوث FOIT. إذا لم يتم تحميل الخط أثناء نافذة الوقت، فسيستخدم المتصفح خطأ احتياطياً.

إحدى الطرق للتعامل مع FOUTs هي استخدام أداة مثل Font Style Matcher للعثور على خط احتياطي أقرب ما يمكن إلى الخط المخصص حتى لا يbedo تغيير الخط جذرًا للغاية. ومع ذلك، يمكننا استخدام خاصية font-display للتعامل مع هذه المشكلات بطريقة أكثر أناقة.

طرق تضمين الخطوط:

Font formats en browsers

bron: <https://transfonter.org/formats>

	Chrome	Firefox	Opera	Safari	Edge
TTF/OTF	4	3.5	10	3.1	9
WOFF	5	3.6	11.5	5.1	9
WOFF2	36	39	23	10	14
SVG	4-37	XX	9-24	3.2	XX
EOT	XX	XX	XX	XX	6-11

WOFF(2)

Web Open Font Format

Created and developed by Mozilla (2009). It is now a W3C Recommendation.

WOFF(2) fonts load faster because they use a more efficient compression algorithm.

WOFF2 for the win and WOFF (and TTF) for fallback.

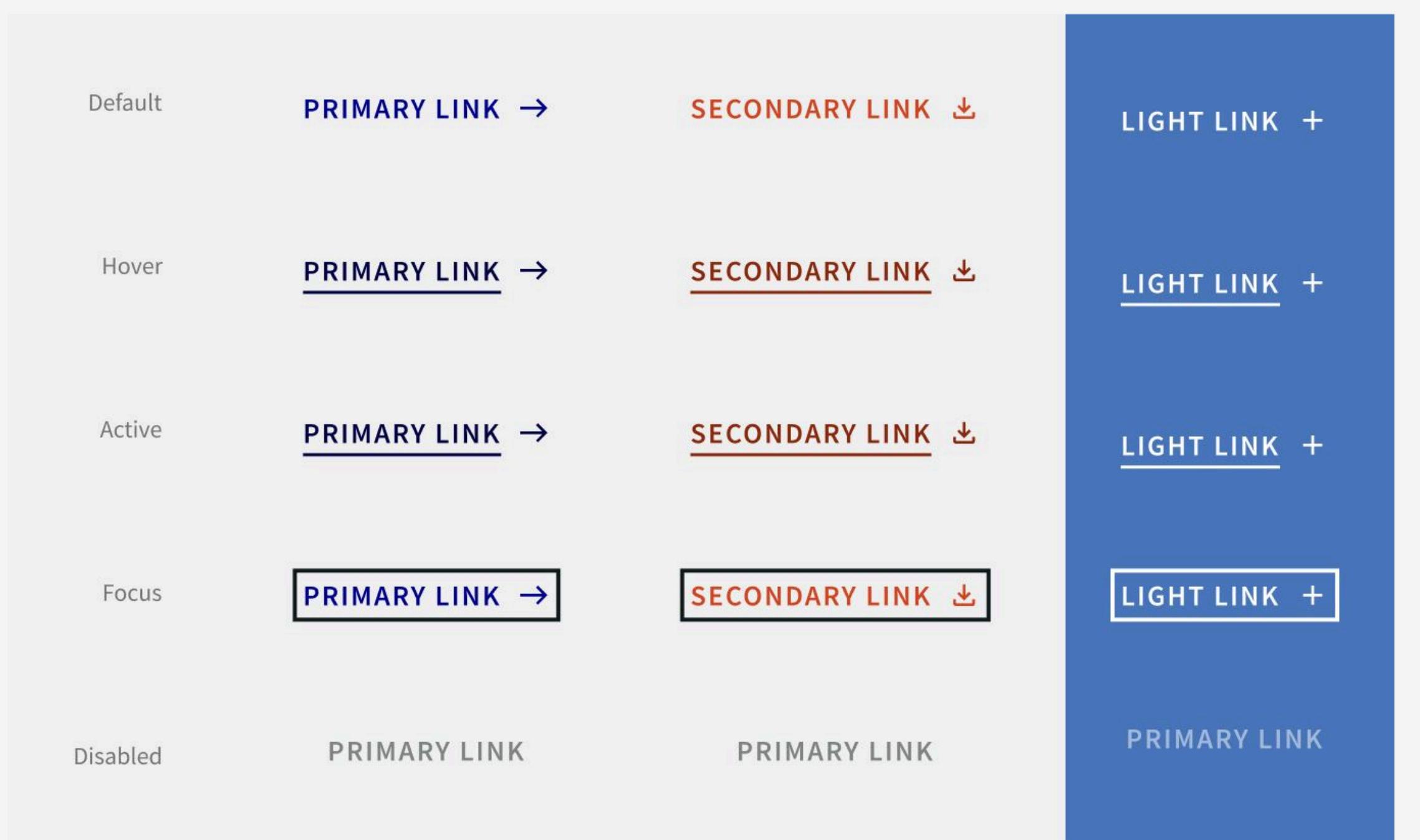
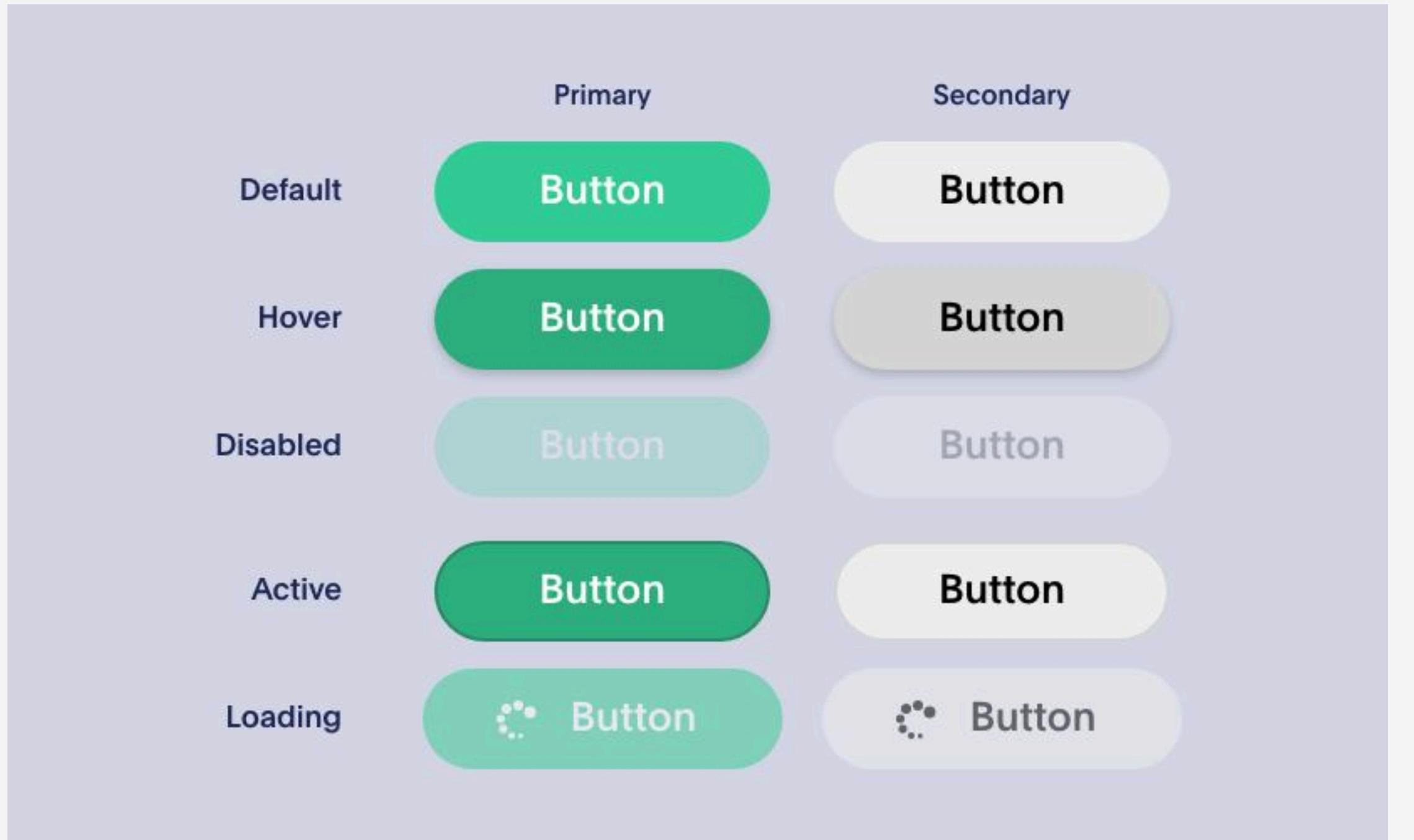
Colors

الالوان

مثال:

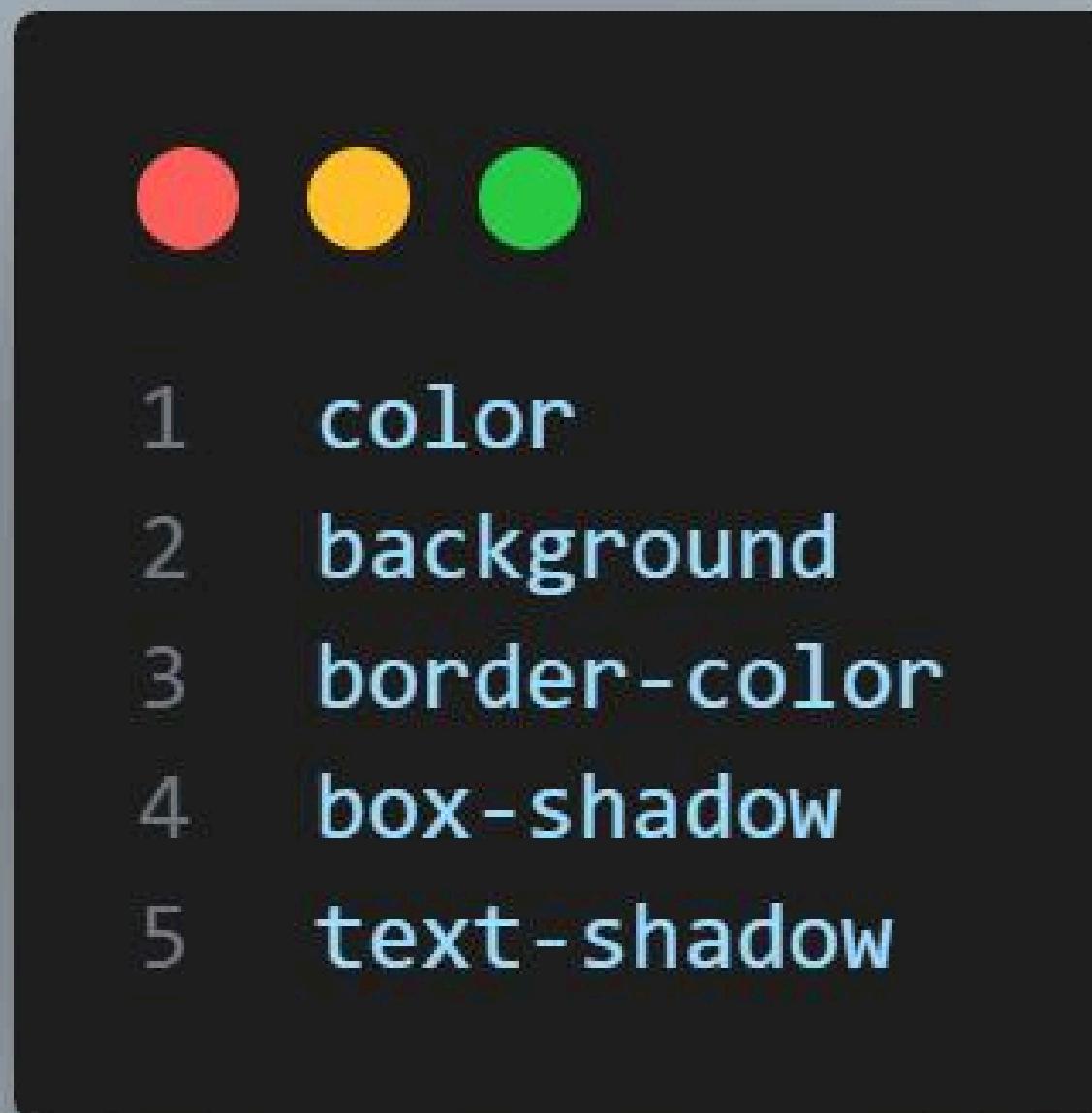
الالوان في الازرار \ الروابط توضح:

- درجة اهمية الزر \ الرابط
- حالة الزر \ الرابط
- حالة العملية



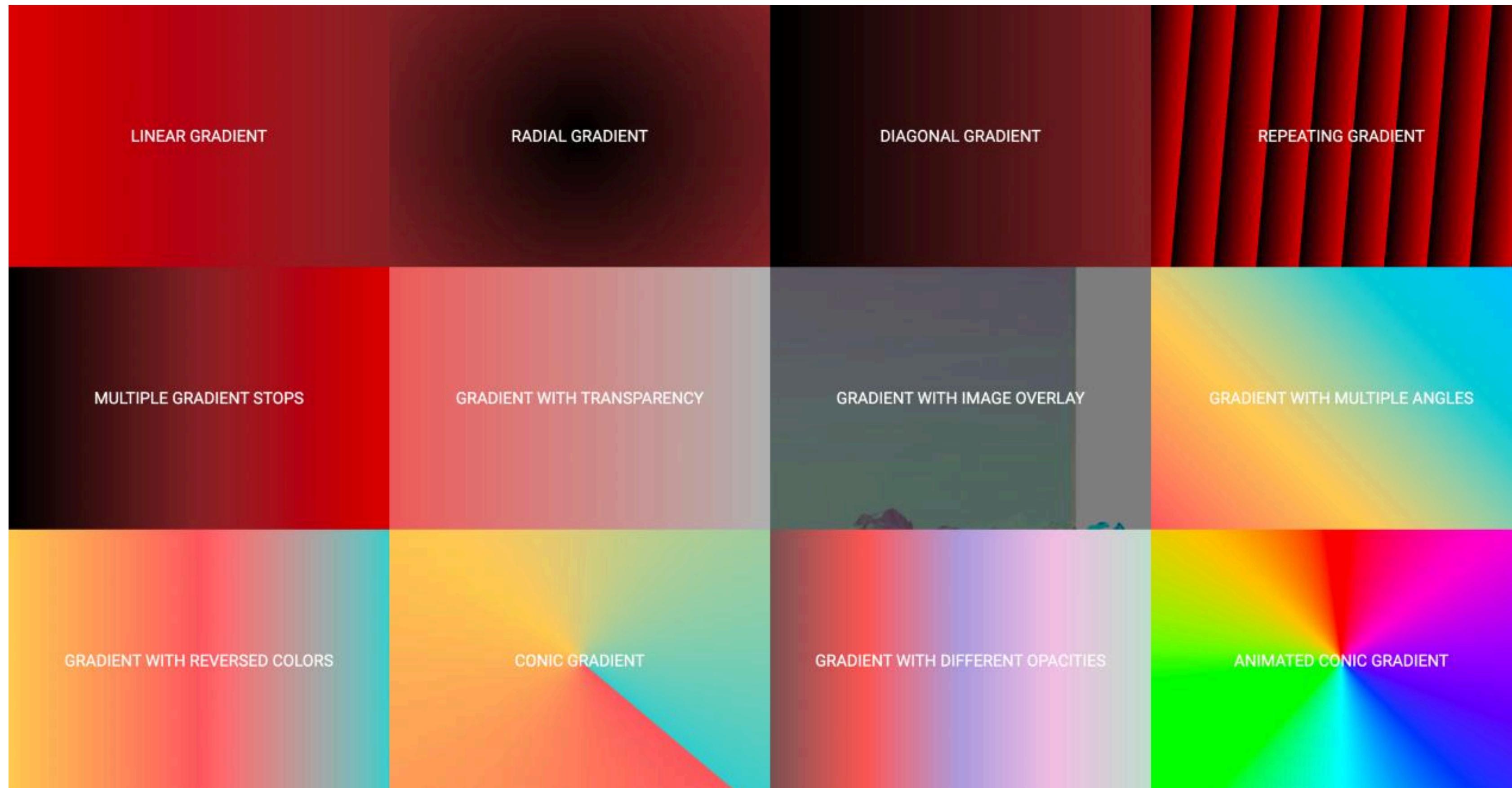
Colors

طرق تحديد الألوان:



- **الألوان بالتسمية** :(Named Colors)
مثال: .red, blue, green
- **القيم الست عشرية** :(Hex Codes)
مثال: #ff5733
- **النظام العددي** :(RGB)
مثال: .rgb(255, 87, 51)
- **النظام العددي مع الشفافية** :(RGBA)
مثال: .rgba(255, 87, 51, 0.8)
- **النظام اللوني** :HSL/HSLA
مثال: .hsla(9, 100%, 60%, 0.8) أو hsl(9, 100%, 60%)

css gradient



[css gradient](#)

اداة تساعد على تنسيق مكونات الالوان يدويا

css gradient

Linear Gradients Using Angles

`background-image: linear-gradient(angle, color-stop1, color-stop2);`



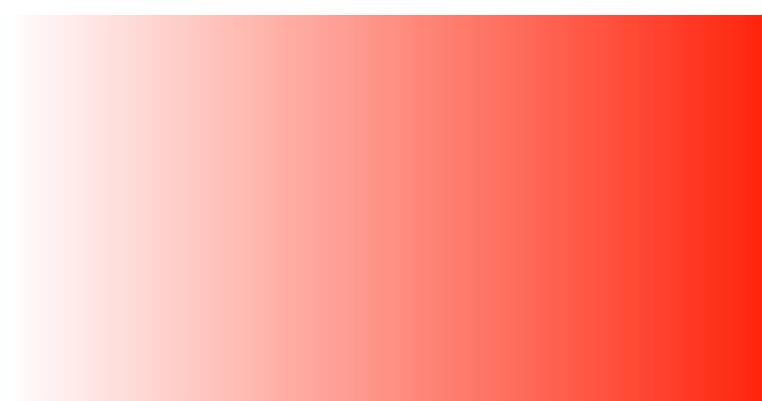
`background-image: linear-gradient(180deg, red, yellow);`

Linear Gradients Using Multiple Color Stops



`background-image: linear-gradient(red, yellow, green);`

Linear Gradients Using Transparency



`background-image: linear-gradient(to right, rgba(255,0,0,0), rgba(255,0,0,1));`

Linear Gradients

`background-image: linear-gradient(direction, color-stop1, color-stop2, ...);`

Direction - Top to Bottom (this is default)



`background-image: linear-gradient(red, yellow);`

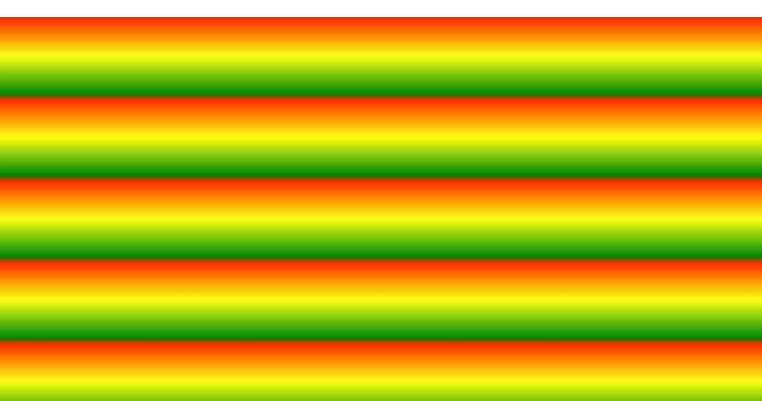


`background-image: linear-gradient(to right, red, yellow);`



`background-image: linear-gradient(to bottom right, red, yellow);`
يمكنك إنشاء تدرج قطريًا عن طريق تحديد مواضع البداية الأفقية والعمودية.

Repeating a linear-gradient



`background-image: repeating-linear-gradient(red, yellow 10%, green 20%);`

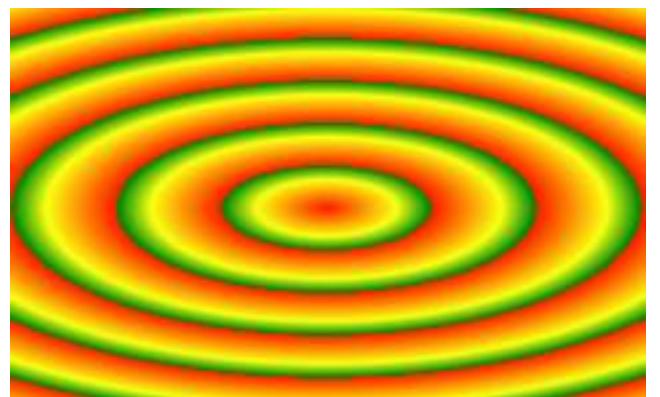


`background-image: linear-gradient(red 0% 30%, yellow 30% 60%, green 100%);`

يمكن تحديد نسب تداخل الألوان بشكل يدوي

css gradient

Repeating a radial-gradient

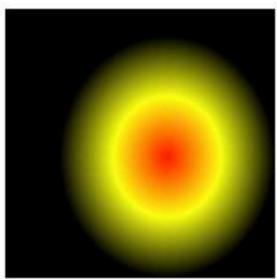


`background-image: repeating-radial-gradient(red, yellow 10%, green 15%);`

CSS Radial Gradients Use of Different Size Keywords

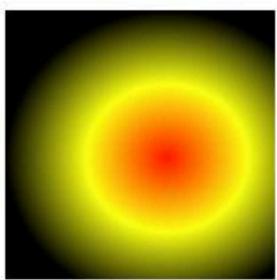
يحدد size parameter حجم التدرج. ويمكن أن يأخذ أربع قيم:

closest-side:



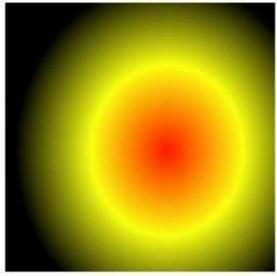
`background-image: radial-gradient(closest-side at 60% 55%, red, yellow, black);`

farthest-side:



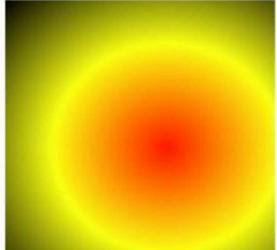
`background-image: radial-gradient(farthest-side at 60% 55%, red, yellow, black);`

closest-corner:



`background-image: radial-gradient(closest-corner at 60% 55%, red, yellow, black);`

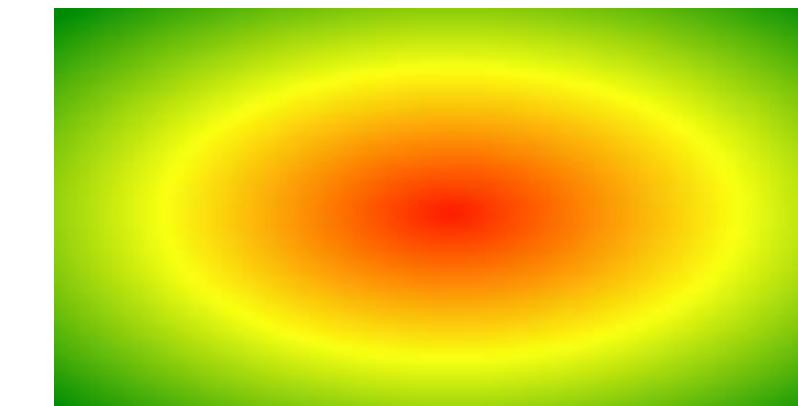
farthest-corner (default):



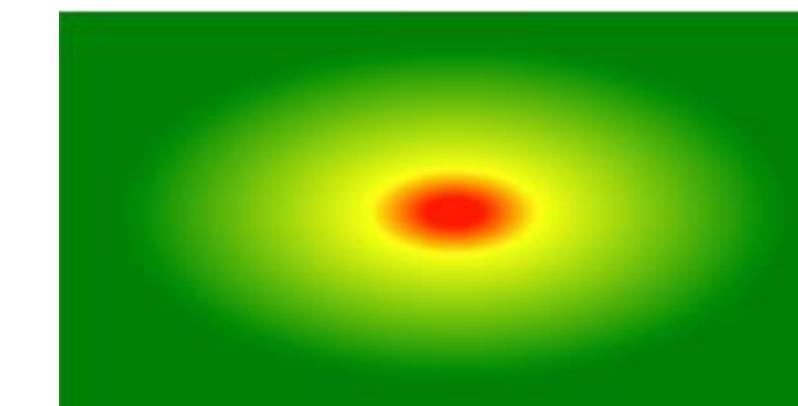
`background-image: radial-gradient(farthest-corner at 60% 55%, red, yellow, black);`

Radial Gradients

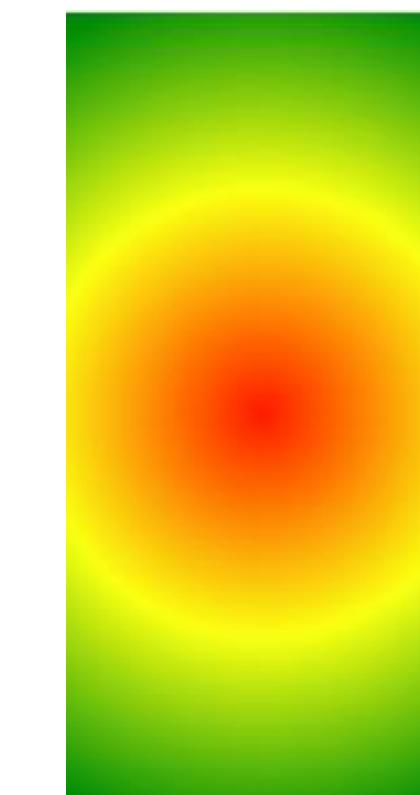
`background-image: radial-gradient(shape size at position, start-color, ..., last-color);`
شكل افتراضي، الشكل هو شكل بيضاوي، والحجم هو أقصى زاوية، والموضع هو المركز.



`background-image: radial-gradient(red, yellow, green);`



`background-image: radial-gradient(red 5%, yellow 15%, green 60%);`

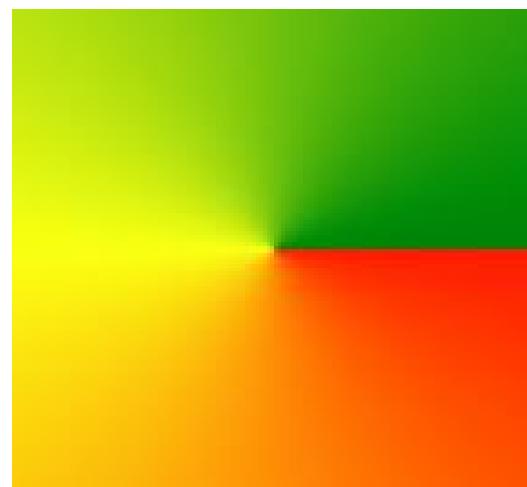


`background-image: radial-gradient(circle, red, yellow, green);`
مهما كان شكل الحاوية ستبقى الألوان تأخذ شكل دائري

css gradient

Conic Gradient With Specified From Angle

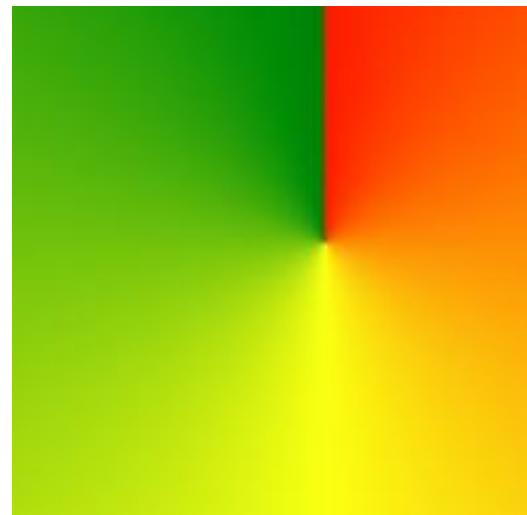
تحدد [from angle] الزاوية التي يدور بها التدرج المخروطي بأكمله.



```
background-image: conic-gradient(from 90deg, red, yellow, green);
```

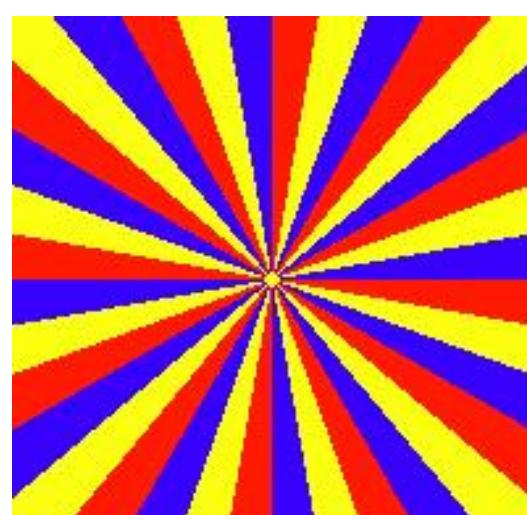
Conic Gradient With Specified Center Position

يحدد [at position] مركز التدرج المخروطي.



```
background-image: conic-gradient(at 60% 45%, red, yellow, green);
```

Repeating a Conic Gradient



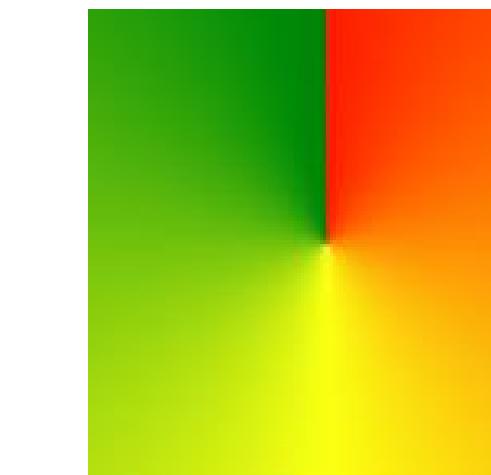
```
background-image: repeating-conic-gradient(
  red 0deg 10deg,
  yellow 10deg 20deg,
  blue 20deg 30deg
);
```

Conic Gradients

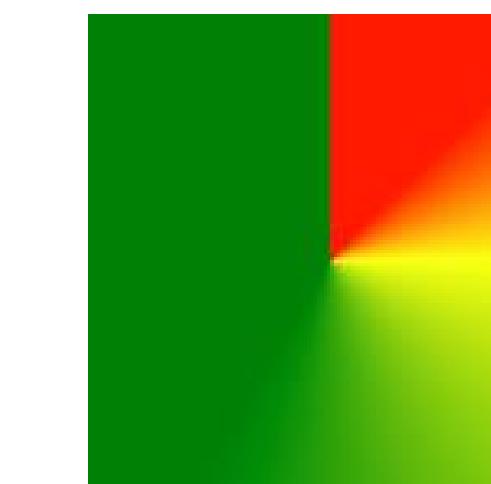
```
background-image: conic-gradient([from angle] [at position,] color [degree],
  color [degree], ...);
```

بشكل افتراضي، الزاوية هي 0 درجة والموضع هو المركز.

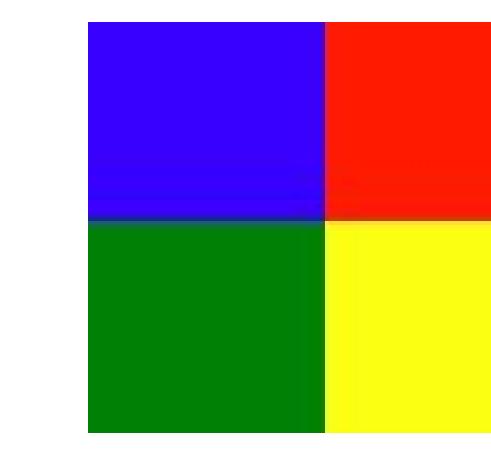
إذا لم يتم تحديد درجة ، فسيتم توزيع الألوان بالتساوي حول نقطة المركز.



```
background-image: conic-gradient(red, yellow, green);
```



```
background-image: conic-gradient(
  red 45deg,
  yellow 90deg,
  green 210deg
);
```



```
background-image: conic-gradient(
  red 0deg 90deg,
  yellow 90deg 180deg,
  green 180deg 270deg,
  blue 270deg
);
```



Custom properties

وهي القدرة على استخدام المتغيرات (variables) في ملف CSS

حيث يتم إعطاء القيمة (value) اسم محدد ليتم استخدامه عوضاً عنها كقيمة

فائدة:

- تسهل عملية استخدام القيم وتعديلها
- تسهل تعديل القيم عند الحاجة
- يمكن استخدامها من أجل القدرة على جعل التنسيق أكثر ديناميكية عند الحاجة لذلك



```

1  :root {
2      /* يتم استخدام -- قبل الاسم لتمييزه كمتغير */
3      --white: #ffffff;
4      --blue: #0000ff;
5
6      /* يمكن استخدام الحروف الكبيرة بدل المسافة */
7      --textColor: var(--white);
8
9      /* يمكن استخدام - بدل المسافة */
10     --primary-color: var(--blue);
11 }
12
13 section {
14     background-color: var(--primary-color);
15 }
16
17 p {
18     color: var(--textColor);
19 }
20

```

Custom properties

عادة يتم وضع المتغيرات داخل “`:root`” حيث يوضع بأعلى ملف الـ CSS
عندما تكون المتغيرات صالحة للاستخدام لكل عناصر المحتوى.

طريقة الإنشاء:

القيمة : اسم المتغير--

طريقة الاستخدام:

اسم المتغير--(القيمة)

أو : (قيمة احتياطية, اسم المتغير-- القيمة)

ملاحظة مهمة: بما ان الـ custom properties تعمل ايضا بشكل عائلي فيمكن ان تكون موجودة داخل الـ body في الـ CSS لكن يتم استخدام الـ `:root` بشكل افتراضي.



```

1  ul {
2    --list-item-color: rgb(184, 7, 184);
3
4    background-color: beige;
5  }
6
7  ul li {
8    color: var(--list-item-color);
9  }
10
11 p {
12   color: var(--list-item-color);
13 }
```

Custom properties

تعمل الـ custom properties بشكل عائلي فقط بما يعنيه:

عند وضع custom propertie داخل حاوية فإنها تصبح صالحة للاستخدام

داخل جميع افراد عائلته العنصر , ابنه, حفيده ... الخ



```

1 <body>
2   <ul>
3     <li>I am beige colored text</li>
4   </ul>
5   <p>I am "p" text</p>
6 </body>
```

I am beige colored text

I am "p" text

ملاحظة مهمة: عند وضع custom propertie داخل عنصر فلا يمكن استخدامها داخل عنصر اخر ليس له قرابة عائلية معه.

Custom properties

- يمكن استخدام قيم مختلفة لـ variables وليس فقط الالوان. مثال:

```
body {  
    --brand-color: #990000;  
    --transparent-black: rgba(0, 0, 0, 0.5);  
  
    --spacing: 0.66rem;  
    --max-reading-length: 70ch;  
    --brandAngle: 22deg;  
  
    --visibility: hidden;  
    --my-name: "Chris Coyier";  
}
```

- :custom properties
- الوضع المظلم
- اضافة ثيم معين (هالوين, كريسمس, الخ)
- رفع امكانيات الـ CSS لمستوى اعلى كتنسيقات او انميشنز .. الخ

قراءة المزيد حول الـ : custom properties
<https://css-tricks.com/a-complete-guide-to-custom-properties/>

The end of CSS les 2