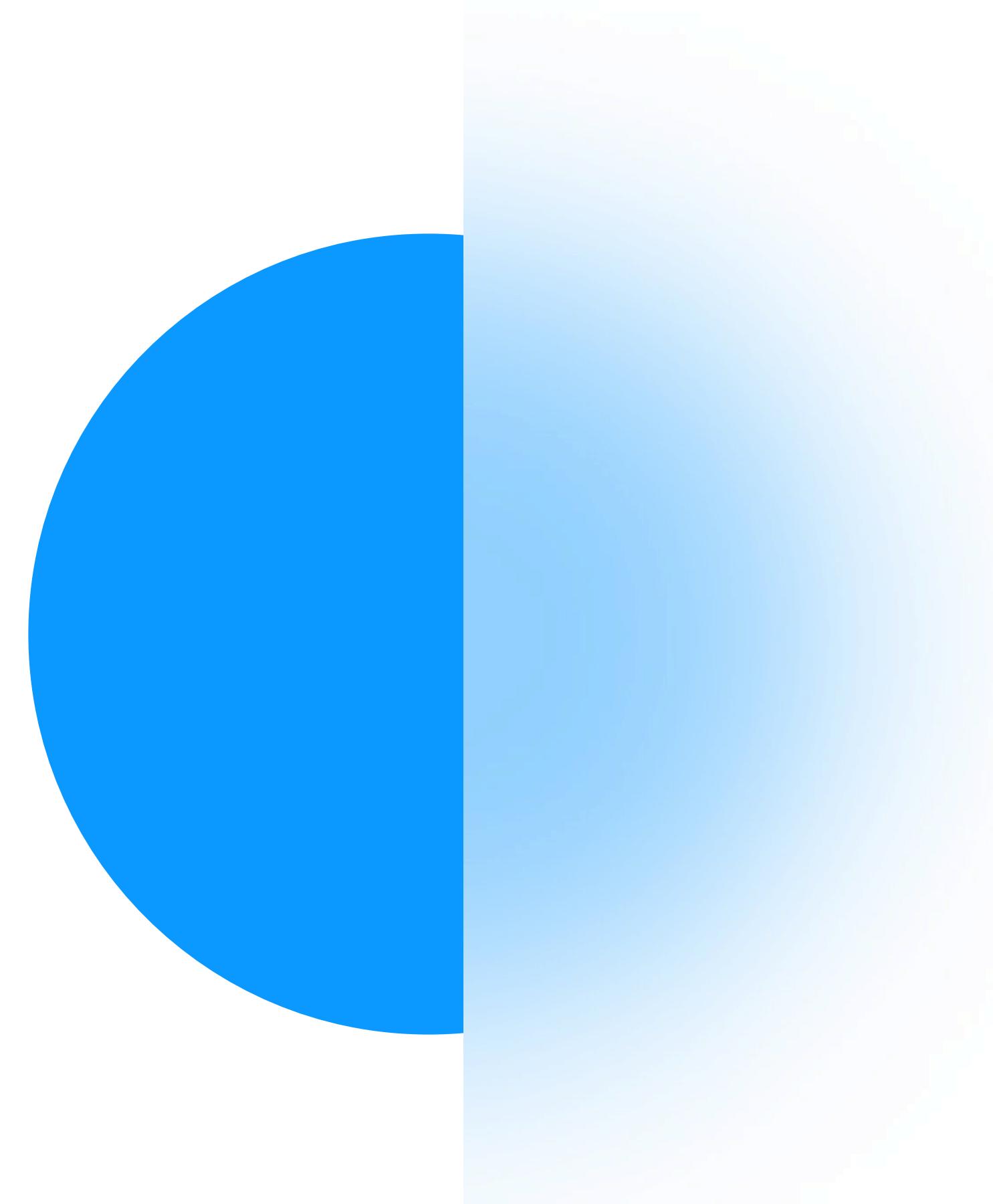


# HTML (HyperText Markup Language)

Les 1



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

# الهيكل الأساسي لملف الـ HTML

```

1 <!DOCTYPE html>
2 <!-- يخبر المتصفح بنوع الملف المستخدم --!>
3
4 <html lang="en">
5   <!-- تعریف اللغة الافتراضية للصفحة/الموقع --!>
6   <head>
7     <!-- يحتوي على معلومات الصفحة غير المرئية --!>
8     <!-- مثل العنوان، التعريف بالملفات الخارجية، الميتادات، إلخ --!>
9     <meta charset="UTF-8" />
10    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
11    <title>Document</title>
12    <!-- عنوان الصفحة --!>
13  </head>
14
15  <body></body>
16  <!-- يحتوي على محتوى الصفحة الرئيسية --!>
17 </html>
18

```

حيث ان وجود الملف الرئيسي هو الاكثر اهمية مع اسم ثابت وهو (index.html) لكي يستطيع التعرف عليه وفتحه.

**ملاحظة مهمة:** جميع ملفات الموقع يجب ان تتبع القواعد التالية:

- جميع ملفات و مجلدات الموقع توضع داخل مجلد واحد.
- عدم وضع ملف الـ index.html داخل مجلد آخر عند بناء موقع بالأساسيات.
- جميع اسماء الملفات تبدأ بحرف صغير.
- عدم وجود مسافات في اسم الملف يمكن استخدام إما - (menu-icon) او حرف كبير مثل (menulcon) للتسمية باكتر من كلمة واحدة.

```

1  <header>
2      <!-- حاوية للمحتوى التمهيدي كقائمة روابط التنقل، اللوغو... الخ --!>
3  <nav>
4      <!-- تحديد قائمة روابط --!>
5  </nav>
6 </header>
7
8 <main>
9     <!-- حاوية للمحتوى الرئيسي للصفحة --!>
10 <article>
11     <!-- تحديد محتوى مستقل يمكن أن يكون مستقلاً عن الصفحة الرئيسية --!>
12 </article>
13
14 <section>
15     <!-- تحديد قسم من المحتوى --!>
16 </section>
17
18 <aside>
19     <!-- تحديد محتوى جانبي غير أساسى للصفحة --!>
20 </aside>
21
22 <time>
23     <!-- تحديد تاريخ أو وقت معين --!>
24 </time>
25
26 <figure>
27     <!-- حاوية تشير إلى محتوى مستقل --!>
28     ، حاوية تشير إلى محتوى مستقل
29     -- مثل الرسوم التوضيحية، والمخططات، والصور، وقوائم الرموز، وما إلى ذلك
30
31 <figcaption>
32     <!-- توضيح للمحتوى الموجود داخل العنصر الأب --!>
33 </figcaption>
34 </figure>
35
36 <details>
37     <!-- تحديد تفاصيل إضافية يمكن للمستخدم فتحها وإغلاقها عند الطلب --!>
38 <summary>
39     <!-- عنوان التفاصيل --!>
40 </summary>
41 </details>
42 </main>
43
44 <footer>
45     <!-- تحديد تذليل الصفحة --!>
46 </footer>
47

```

# العناصر الدلالية

## Semantic Elements

تحسين من تنظيم المحتوى وتساعد محركات البحث على فهم الصفحة.

**ملاحظة مهمة:** جميع ملفات الموقع يجب ان تتبع القواعد التالية:

- جميع ملفات و مجلدات الموقع توضع داخل مجلد واحد.
- عدم وضع ملف الـ index.html داخل مجلد آخر عند بناء موقع بالأساسيات.
- جميع اسماء الملفات تبدأ بحرف صغير.
- عدم وجود مسافات في اسم الملف يمكن استخدام إما - (menu-icon) او حرف كبير مثل (menulcon) للتسمية باكتر من كلمة واحدة.

# العناصر غير الدلالية

## Non-semantic elements

حيث ان وجود الملف الرئيسي هو الاكثر اهمية مع اسم ثابت وهو (index.html) لكي يستطيع التعرف عليه وفتحه.

**ملاحظة مهمة:** بعض العناصر لا تقبل إلا ابناء محددين كمثال:

- كالقوائم: <table>, <tr>, <ul>, <ol>
- عناصر النصوص <act>, <p> ..

```

1  -- العنانيون والفقريات --!>
2  -- لعنانيون --!>
3  <h1></h1>
4  -- حتى --!>
5  <h6></h6>
6
7  -- للفكريات --!>
8  <!-- paragraph -->
9  <p></p>
10
11 -- القوائم --!>
12 <nav></nav>
13 -- للقوائم الرئيسية (قوائم التنقل في الموقع) --!>
14
15 <aside></aside>
16 -- للقوائم الجانبية --!>
17
18 <ul>
19   -- قائمة غير مرتبة
20   <li></li>
21   <!-- list item -->
22 </ul>
23 <ol>
24   -- قائمة مرتبة
25   <li></li>
26   <!-- list item -->
27 </ol>
28
29 -- الروابط --!>
30 <a href=""></a>
31 <!-- hyperlink: -->
32 -- ارتباط تشعبي أي ربط المفحات
33
34 -- الصور --!>
35 <img src="" alt="" />
36 <!-- الصوت --!>
37 <audio src="" controls></audio>
38
39 -- الفيديو --!>
40 <video src="" controls></video>
41
42 -- الأزرار --!>
43 <button></button>
44
45

```

```

1  -- الجداول --!>
2  <table>
3    <tr>
4      <!-- table row -->
5      <th></th>
6      <!-- table header -->
7    </tr>
8    <tr>
9      <!-- table row -->
10     <td></td>
11     <!-- table data -->
12   </tr>
13 </table>
14
15 -- النماذج --!>
16 <form action="">
17   <label for=""></label>
18   <!-- للتعليق على العنصر -->
19   <input type="" id="" />
20   <!-- لإدخال البيانات -->
21   <output name="" for=""></output>
22   <!-- لعرض نتيجة عملية -->
23   <textarea name="" id=""></textarea>
24   <!-- لإدخال النصوص الطويلة -->
25   <select name="" id="">
26     <!-- لإدخال القوائم المنسدلة -->
27     <option value=""></option>
28     <!-- لإدخال الخيارات -->
29   </select>
30   <fieldset></fieldset>
31   <!-- لجمع قسم من العناصر في مجموعة واحدة -->
32
33   <input type="submit" value="" />
34   <!-- لإرسال النموذج -->
35   <input type="reset" value="" />
36   <!-- لإعادة تعيين النموذج -->
37 </form>

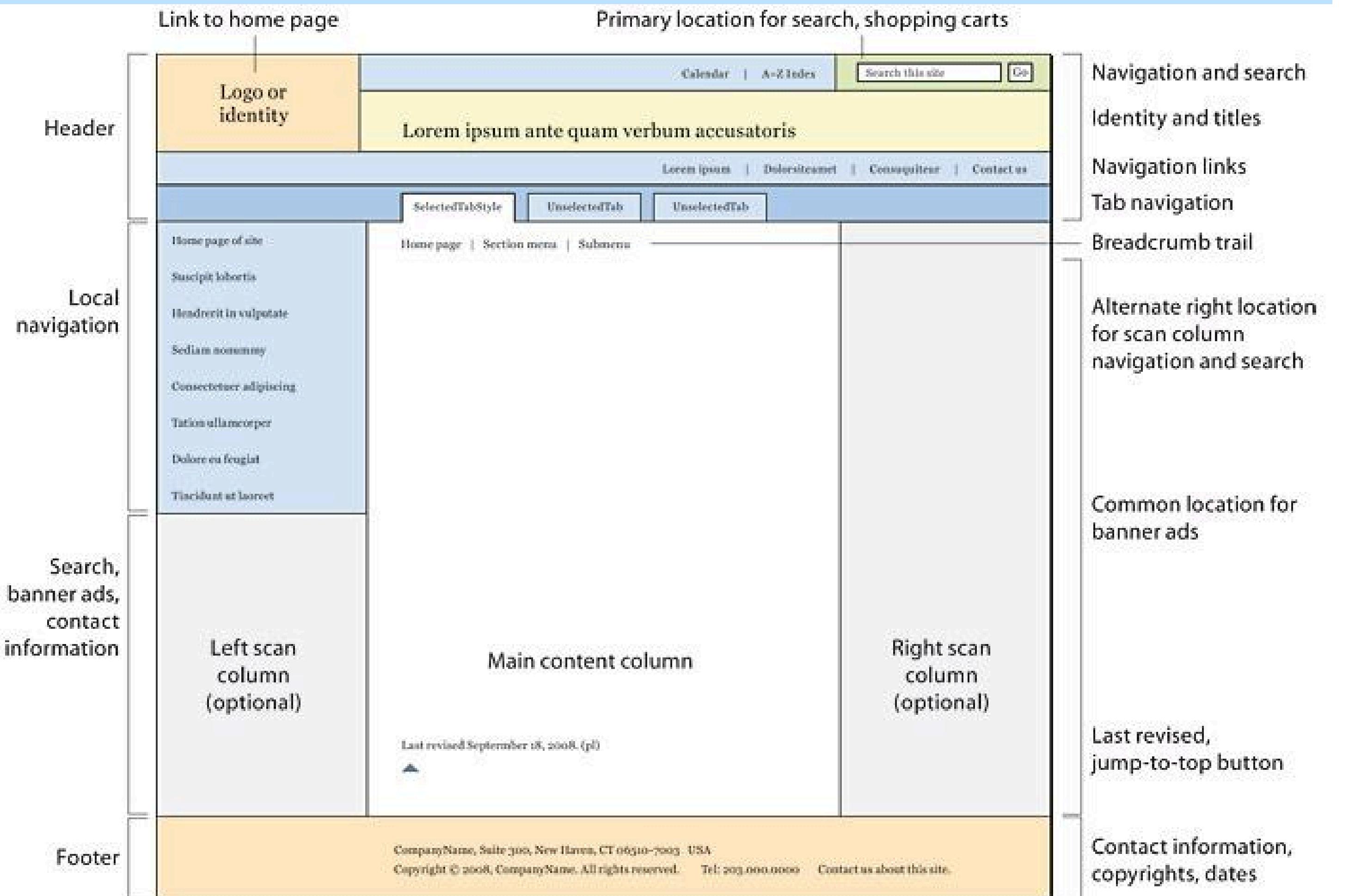
```

# تقسيم المحتوى

في المثال الجانبي أحد أمثلة تقسيم محتوى الصفحة

داخل الكود

**ملاحظة مهمة:** تختلف طريقة تقسيم المحتوى حسب التصميم



```
1 <body>
2   <!-- this element is: grand parent -->
3   <ul>
4     <!-- this element is: parent -->
5     <li>
6       <!-- this element is: child -->
7     </li>
8   </ul>
9 </body>
```

هذا المثال يوضح وصف العلاقات الهيكلية بين العناصر داخل شجرة DOM (Document Object Model) حيث ان الـ `ul` هي الابن المباشر للـ `body` و الـ `li` هي الابن المباشر للـ `ul` و حفيد للـ `body`.



هذه العلاقات تُساعد على تنظيم وتنسيق المحتوى في صفحة الويب، وتستخدم بشكل كبير عند تطبيق أنماط CSS أو التعامل مع عناصر الصفحة باستخدام JavaScript.

#### ملاحظة مهمة:

- العنصر الأب: العنصر الذي يحتوي على عناصر أخرى.
- العنصر الابن: العنصر الذي يقع مباشرةً داخل عنصر آخر.
- العنصر الحفيد: أي عنصر موجود داخل عنصر آخر، سواءً كان بشكل مباشر أو من خلال تداخل متعدد المستويات.

# العناصر غير الدلالية

## Non-semantic elements

- **div**

1. عنصر كتلي (Block-level element).
2. يُستخدم لتجميع أقسام كبيرة أو عناصر متعددة معاً.
3. يشغل العرض الكامل للعنصر الأب، مما يجعله يبدأ دائمًا من سطر جديد.
4. يُستخدم عادةً مع السمات مثل class أو id لتطبيق تنسيقات CSS أو وظائف JavaScript.

- **span**

1. عنصر سطري (Inline element).
2. يُستخدم لتجميع أجزاء صغيرة من النص أو المحتوى داخل سطر واحد.
3. لا يسبب كسر السطر، بل يُبقي العناصر متجاورة مع باقي المحتوى.
4. يُستخدم أيضًا لتطبيق تنسيقات محددة على جزء معين من النص أو لتحديده بواسطة class أو id.
5. يمكن تحويله إلى Block element بإستخدام الـ CSS

باختصار، تُستخدم كل من <div> و <span> كأدوات للتقسيم والتنظيم داخل صفحة HTML دون إضافة معنى دلالي للمحتوى، ويمكن تمييزها واستخدامها بكفاءة عند الحاجة إلى تصميم تخطيطات معينة أو تطبيق أنماط محددة على أجزاء معينة من المحتوى.

**ملاحظة مهمة:** من الأفضل للمبتدئين استخدامهم فقط لهدف تنسيقي أو جمالي

# الفرق بين

# Inline Elements & Block-Level Elements

Inline Elements	Block-Level Elements	الميزة
تبقى في نفس السطر	تبدأ على سطر جديد	السطر
عرضها يعتمد على المحتوى فقط	تمتد لملئ عرض الحاوية الاب	العرض
يمكن أن تحتوي فقط على inline	يمكن أن تحتوي على عناصر block أو inline	الاحتواء
تنسيق النصوص أو إضافة محتويات صغيرة	هيكلة المحتوى	الاستخدام

## أهم السمات التي يجب تمييزها:

لتحديد لغة الصفحة / الموقع	<b>lang</b>
لتحديد هوية العنصر ولا يمكن استخدام نفس الهوية أكثر من مرة.	<b>id</b>
لتحديد اسم للعنصر ويمكن ان يستخدم اكثر من مرة. ينصح استخدامه فقط عند الحاجة إليه.	<b>class</b>
يحدد عنوان URL لملف الوسائط. <code>&lt;audio&gt;, &lt;embed&gt;, &lt;iframe&gt;, &lt;img&gt;, &lt;input&gt;, &lt;script&gt;, &lt;video&gt;</code>	<b>src</b>
يحدد نصاً بديلاً عندما يفشل العنصر الأصلي في العرض. <code>&lt;area&gt;, &lt;img&gt;, &lt;input&gt;</code>	<b>alt</b>
يحدد عنوان URL للصفحة التي ينتقل إليها الرابط.	<b>href</b>
يحدد نوع العنصر	<b>type</b>
يحدد نمط CSS المضمن لعنصر	<b>style</b>
يحدد قيمة العنصر. <code>&lt;input&gt;, &lt;option&gt;</code>	<b>value</b>
يحدد الهدف لمكان فتح المستند المرتبط أو مكان إرسال النموذج. نستخدم لها القمة <code>_blank</code> لفتح الرابط بصفحة جديدة.	<b>target</b>
يتم استخدامه داخل العنصر <code>&lt;time&gt;</code> لترجمة الوقت إلى تنسيق قابل للقراءة عن طريق قارئ الشاشة مثلاً أو يتم ربطه بأمور أخرى أكثر تعقيداً كالتوقيع الخاص بالمستخدم.	<b>datetime</b>

# السمات (Attributes)

تُستخدم لتوفير معلومات إضافية عن العناصر. منها يجب وجوده و ملئه و أخرى حسب الرغبة او الحاجة.

**ملاحظة مهمة:** يمكن استخدامها لجعل التنسيقات داخل الـ css أكثر كفاءة أبضاً.

[HTML checker](#) موقع يسمح بالتحقق من كود الـ html واكتشاف الأخطاء

**ملاحظة مهمة:** يمكن التركيز فقط على الإيورز و تجاهل التحذيرات (Warnings)

## HTML Input Types

<input type="button">	Button
<input type="checkbox">	<input checked="" type="checkbox"/>
<input type="color">	<input type="color" value="red"/>
<input type="date">	dd-mm-yyyy <input type="button" value="calendar"/>
<input type="email">	farazc60@gmail.com
<input type="file">	Choose File No file chosen
<input type="hidden">	
<input type="image">	
<input type="number">	5 <input type="button" value="up"/>
<input type="password">	*****
<input type="radio">	<input checked="" type="radio"/> <input type="radio"/>
<input type="range">	<input type="range" value="50"/>
<input type="reset">	Reset
<input type="submit">	Submit
<input type="text">	codewithfaraz
<input type="url">	<input type="text" value="https://www.codewithfaraz.com"/>

# انواع الـ input types

يحدد الـ type كيفية تعامل الكود و المستخدم على حد سواء مع خانة الإدخال.



## PROFILE

## YOASOBI

コンポーザーのAyase、ボーカルのikuraからなる「小説を音楽にするユニット」。2019年11月に公開された第一弾楽曲「夜に駆ける」は2020年1月にストリーミング再生3億回を突破し、Billboard Japan 総合ソングチャート「HOT100」にて2020年年間1位を獲得。第二弾楽曲「あの夢をなぞって」は原作小説がコミカライズ、第三弾楽曲「ハーリジョン」は飲料や映像作品とのコラボレーションを果たし、7月20日に第四弾楽曲「たぶん」、9月1日にブルーイング「アルフォートミーチョコレート」CMソング「群青」をリリース。12月には鈴木おさむが原作小説を手掛けた楽曲「ULTRA」を発表。2021年1月6日に満を持して初のCD「THE BOOK」をリリース。豪華な仕様が大いに話題を呼び、オリコンデイリーランキング初登場2位を記録しながら、配信では全ての楽曲がApple Musicストリーミングチャートで1位以内に同時ランクインするという快挙を達成した。原作小説の書籍化や映画化など、音楽以外の領域にも展開の幅を広げている。



Ayase



VOCALOID producer / YOASOBI composer

1994年4月4日生まれ、山口県出身。2018年12月にVOCALOID楽曲を投稿開始。切なさと哀愁を帯びたメロディ、考察意欲を描き立てる歌詞で人気を博し、2019年4月に発表した「ラストリゾート」はYouTube1300万再生突破。2019年11月リリースの初EP「幽靈東京」は即完。ボーカル楽曲を自身が歌唱するセレブカバーにも定評があり、「幽靈東京」は1900万、「夜撫でるメノウ」も2200万再生を突破。ボーカロイド、YOASOBIのコンポーザーとしての活動に加えさまざまなアーティストへの楽曲提供も手掛ける。

## ikura



Singer - Songwriter / YOASOBI vocal

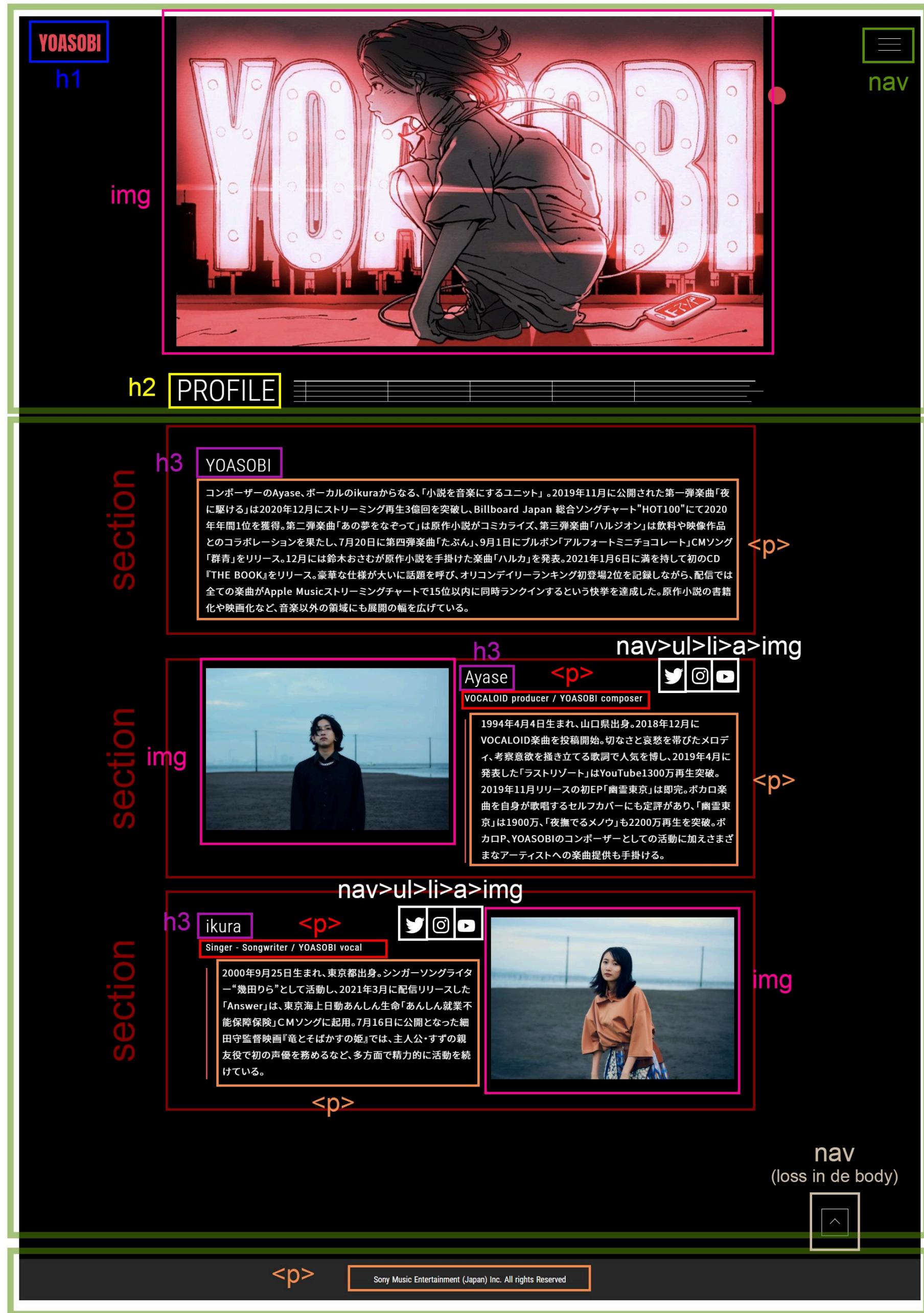


2000年9月25日生まれ、東京都出身。シンガーソングライターとして活動し、2021年3月に配信リリースした「Answer」は、東京海上日動あんしん生命「あんしん就業不能保障保険」CMソングに起用。7月16日に公開となった細田守監督映画『君とそばかすの姫』では、主人公・すずの親友役で初の声優を務めるなど、多方面で精力的に活動を続けている。

لبناء هيكل html صحيح من البداية يجب علينا دراسة التصميم  
المقدم لنا بهدوء.  
حيث ننظر إليه من جانب هيكتي كمحتوى وليس جمالي كتصميم.

# تشريح التصميم

header



# تشريح التصميم

يبين المثال كيفية تحليل التصميم مما يسهل علينا تحويله الى كود html بسهولة.

حيث من الواضح ما هي العناصر التي سيتم استخدامها و كيفية هيكلتها.

**ملاحظة مهمة:** الكتابة `nav>ul>li` هي طريقة لذكر العناصر حسب ترتيبها داخل بعضها، حيث أن "`>`" تعني الابن المباشر.

**ملاحظة مهمة:** العناصر التي ليست بمحتوى فعال وإنما جمالى فقط فيتم اضافتها عن طريق الـ CSS، كمثال: الخط الاحمر العاومودي بجانب النصوص. او بإستخدام عنصر مساعد في الـ `html`, كمثال مجموعة الخطوط اسفل الصورة حيث يمكن استخدام `div` كحاوية بداخلها `span` لكل خط.

The end of HTML les 1