

# Html And Css Exam

## 1- Explain :has() pseudo-class

دي بيسموها "Parent Selector" علشان بحددها عن طريق اللي جواها يعني مثلاً

```
body>  
<div>  
  <img src="" alt="">  
</div>
```

هنا لو عايز اختار أي div جواه صورة بنعمل كده

```
div:has(img) {  
  |  
}
```

كده بيحب كل ال divs اللي جواهم صورة

## 2- What is CSS aspect-ratio property?

ال aspect-ratio دي خاصية بتخلينا نثبت النسبة بين العرض والطول للعنصر. أهميتها بتبان جداً في ال Responsive Design؛ لأنها بتضمن إن الصور أو الفيديوها متضغطش (stretch) لما حجم الشاشة يتغير، وبتحافظ على ال UI متناسق من غير ما نضطر نحدد Height ثابت لكل شاشة

```
.video-container {  
  width: 100%;  
  aspect-ratio: 16 / 9;  
}
```

## 3- How would you optimize CSS for performance?

**Minification:** أشيل المسافات والتعليقات.

**Unused CSS:** أمسح الكود اللي مش مستخدم (بأدوات زي PurgeCSS)

**Critical CSS:** أحمل الكود المهم لأول جزء بيظهر للمستخدم (Above the fold) في ال <head>.

**Avoid @import:** لأنها بتعمل تسلسل في التحميل ببطء الدنيا.

## 4- CSS Custom Properties (Variables) vs Sass variables?

- بتقدر تغير قيمتها بال JavaScript في وقت التشغيل (Runtime)
- بتقدر تغير قيمتها جوه Media Query (مثلاً لون ال background يتغير في الموبايل بسطر واحد)
- بينا Sass variables هي "Static"، بمجرد ما الكود بيتحول لـ CSS عادي قيمتها بتثبت وما ينفعش تتغير في المتصفح.

## 5- Grid Calculation (500px container)

- النتيجة: العمود الأول 50px، الثاني 150px، الثالث 300px

## 6- What is the Critical Rendering Path (CRP)?

هي الخطوات التي المتصفح يمشي فيها من أول ما يستلم كود ال HTML لحد ما يرسم البيكسلات على الشاشة. بتعدي بـ 5 مراحل:

1. بناء ال DOM (Document Object Model): المتصفح يقرأ كود ال HTML ويحوله لـ "شجرة من العناصر" (Nodes)

دي المرحلة اللي المتصفح يفهم فيها هيكل الصفحة (مين الأب ومين الابن).

2. بناء ال CSSOM (CSS Object Model): المتصفح بيحلل ملفات ال CSS وقواعد التنسيق. المرحلة دي مهمة عشان

المتصفح يعرف كل عنصر ستايله إيه، وإيه الخصائص اللي هيورثها من العناصر اللي فوقيه.

3. بناء ال Render Tree: هنا المتصفح يدمج ال DOM مع ال CSSOM النتيجة بتكون شجرة ثانية خالص فيها بس العناصر

اللي "هتظهر" فعلاً للمستخدم (يعني أي حاجة واحدة `display: none` بتتمسح من الشجرة دي)

4. مرحلة ال Layout (أو Reflow): بعد ما عرفنا إيه اللي هيظهر، المتصفح بيبدأ يحسب "الأرقام". بيحدد بالظبط كل عنصر

مكانه فين (Coordinates) ومساحته كام بيكسل بالنسبة لحجم شاشة المستخدم (Viewport).

○ ملحوظة: أي تغيير في عرض الشاشة بيخلي المتصفح يعيد الخطوة دي.

5. مرحلة ال Paint: دي آخر مرحلة، وهي "التلوين". المتصفح بيمسك كل الحسابات اللي عملها ويحولها لبيكسلات حقيقية

(Pixels) على الشاشة. هنا بيترسم النص، الألوان، الصور، الظلال (Shadows)، والحواف (Borders)

## 7- What is the purpose of clamp()?

دي بتعمل "Responsive Typography" أو أحجام مرنة من غير Media Queries بتاخذ 3 قيم: (أقل حجم، الحجم المثالي، أكبر حجم).

• مثال: `font-size: clamp(1rem, 5vw, 2rem)`

• ده معناه الخط هيكبر ويصغر مع الشاشة بس مش هيقبل عن 1rem ولا هيزيد عن 2rem

## 8- What is 'Layout Thrashing'?

دي مشكلة أداء بتحصل لما ال JavaScript يقعد يقرأ ويعدل في ال DOM ورا بعضه بسرعة (زي إنك تجيب offsetHeight وبعدين تغير style.height جوه loop ده يجبر المتصفح إنه يعيد حسابات ال Layout كذا مرة في ثانية واحدة، فيخلي الصفحة "تهنج")

## 9- Best way for three choices (Select only one)?

أفضل وأصح طريقة هي استخدام ال (<input type="radio">) Radio Buttons ويكون ليهم نفس ال name attribute ده بيضمن إن المتصفح لوحده يخليك تختار واحد بس، وكرمان هي الطريقة الأصح للـ Accessibility (عشان اللي بيستخدموا Screen Readers)

## 10- :nth-child for every 3rd item starting from 2nd?

المعادلة هتكون:

**:nth-child(3n + 2)**

• الشرح: 3n يعني كل 3 عناصر، و 2 + يعني ابدأ من العنصر الثاني.

## QUESTION 3

### 1- Flex-direction Values:

The flex-direction property defines the main axis along which flex items are placed in the flex container.

- **Example 1:** row (Default direction; items are placed from left to right).
- **Example 2:** row-reverse (Items are placed horizontally but in reverse order).
- **Example 3:** column (Items are placed vertically from top to bottom).
- **Example 4:** column-reverse (Items are placed vertically from bottom to top).

### 2- Blue Elements Selection:

The selector used is `h2 ~ p`, which is the General Sibling Combinator. It selects all `<p>` elements that follow an `<h2>` element within the same parent container.

Analysis of the HTML structure:

- **P1:** Will NOT be blue because it precedes the `<h2>`.
- **H2:** The reference element.
- **P3:** Will be BLUE because it is a sibling that follows the `<h2>`.
- **P4:** Will be BLUE because it is also a sibling that follows the `<h2>`.

