

مشروع عملي

من خلال هاتين المحاضرتين سيتم بناء تخطيط متجاوب لكافة مقاسات الشاشات، بالإضافة إلى تمرير بعض المفاهيم الأساسية الجديدة في CSS مثل استخدام المتغيرات.

1- التقسيم العام وتضمين ملفات CSS الضرورية

```
<!DOCTYPE html>
<html lang="ar" dir="rtl">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>معهد البيرم</title>
  <link rel="stylesheet" href="style.css">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/7.0.0/css/all.min.css" integrity="sha512-
DxV+EoAD0kOygM4IR9yXP8Sb2qwgidEmeqAEmDKIOfPRQZOWbXCzLC6vjbZyy0vPisbH2SyW27+ddLVCN+OMzQ
==" crossorigin="anonymous" referrerpolicy="no-referrer" />
</head>
<body>
  <main>
    <nav></nav>
    <div class="hero"></div>
    <div class="activities"></div>
  </main>
  <footer></footer>
</body>
</html>
```

ملاحظة إنتاجية:

إذا كتبت hero. في vsc فسيعتبر أنك تريد عنصر div من الصف hero، مما يؤكد أهمية هذا العنصر في .html

Ctrl+k+c يفتح ويغلق تعليقاً ويضع مؤشر الماوس جاهزاً للكتابة

استخدام مكتبة fontawesome cdn

ندخل إلى الرابط

<https://cdnjs.com/libraries/font-awesome>

ننسخ الوسم الخاص (copy link tag) بتضمين المكتبة all وهو التالي:

```
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/7.0.0/css/all.min.css" integrity="sha512-DxV+EoAD0kOygM4IR9yXP8Sb2qwgidEmeqAEmDKIOFPRQZOWbXCzLC6vjbZyy0vPisbH2SyW27+ddLVCN+OMzQ==" crossorigin="anonymous" referrerpolicy="no-referrer" />
```

ونقوم بلصقه في جزء ال head.

لدينا الان وصول لكل ايقونات font awesome.

الاعدادات العامة للصفحة:

نضع في الملف style.css القواعد التالية:

```
:root{
  --white:#fff;
  --gray:#c4c4c4;
  --dark-gray:#474b5e;
  --dark:#252731;
  --light-green:#3bd33b;
  --green:#47b847;
}
*{
  margin: 0;
  padding: 0;
  box-sizing: border-box;
}
html{
  font-size: 16px;
}
body{
  background-color: var(--gray);
  font-family: sans-serif;
}
main{
  max-width: 1366px;
  margin: 0 auto;
}
```

القسم nav يتألف من ثلاثة مكونات زر وقائمة وصورة.

```
<nav>
  <button class="cta dark">تسجيل دخول</button>
  <ul class="nav-items">
    <li class="nav-item">
      <a href="#" class="nav-link">الصفحة الرئيسية</a>
    </li>
    <li class="nav-item">
      <a href="#" class="nav-link">من نحن</a>
    </li>
    <li class="nav-item">
      <a href="#" class="nav-link">فعالياتنا</a>
    </li>
  </ul>
  
</nav>
```

أما إعدادات CSS المناسبة لهذا القسم فهي:

```
nav{
  display: flex;
  justify-content: space-around;
  align-items: center;
  height: 12vh;
  background-color: var(--green);
}
.cta{
  font-family: inherit;
  padding: 0.5rem 1rem;
  border: none;
  color: var(--white);
  border-radius: 1rem;
  cursor: pointer;
  font-size: 1rem;
}
.cta.dark{
  background-color: var(--dark);
  transition: background-color 0.3s;
}
.cta.dark:hover{
  background-color: var(--dark-gray);
}
.cta.green{
  background-color: var(--green);
  transition: background-color 0.3sec;
}
.cta.green:hover{
  background-color: var(--light-green);
}
```

```

}
nav .nav-items{
  list-style-type: none;
  display: flex;
}
nav .nav-items .nav-item{
  padding-left: 2rem;
}
.nav-item .nav-link{
  text-decoration: none;
  color:var(--dark);
}
.nav-item .nav-link:hover{
  color:var(--white);
}
nav .logo{
  max-height: 100%;
  border-radius: 50%;
}

```

3- القسم hero

يتألف القسم hero من عنصر **h1** بداخله **span** تستخدم لوسم إحدى كلماته بخصائص معينة، وعنصر **div** يحتوي على زرین كما هو مبين في الكود التالي:

```

<div class="hero">
  <h1>لكافة الدورات التعليمية <span>البيرم</span>معهد</h1>
  <div class="ctas">
    <button class="cta green">سجل لتبدأ رحلتك</button>
    <button class="cta dark">فعالياتنا</button>
  </div>
</div>

```

أما إعدادات CSS المناسبة فهي:

```

.hero{
  height: 40vh;
  display: flex;
  flex-direction: column;
  justify-content: center;
  align-items: center;
}
.hero h1{
  font-size: 3rem;
  width:40%;
}

```

```

text-align: center;
color:var(--dark);
margin-bottom: 2rem;
}
.hero h1 span{
text-shadow: 0 0 10px var(--green);
font-style: italic;
}

```

4- القسم activities

يتألف القسم activities من عنصر h3 وعنصر h6 وعنصر div يمثل قائمة الفعاليات (activities-list).

تضم activities-list مجموعة عناصر div كل منها يمثل فعالية (activity).

تضم كل activity عنصر div يمثل محتوى الفعالية (content) وصورة.

```

<div class="activities">
  <h3>فعالياتنا</h3>
  <h6>ابحث عن شغفك وبادر إلى التسجيل</h6>
  <div class="activities-list">
    <div class="activity">
      <div class="content">
        <h2>دورة الروبوت</h2>
        <p>هل طفلك يحب الابتكار ويهوى استكشاف كيف تعمل الأشياء من حوله؟</p>
        للأطفال من عمر 8 إلى Robotics LEGO Mindstorm EV3 الآن حان الوقت ليخطو أولى خطواته في عالم الروبوتات والبرمجة مع كورس
        . 16 سنة، بأسلوب عملي وتفاعلي يُحَقِّز الإبداع وينمّي مهارات التفكير المنطقي مع المهندسة عبير حسن
      </p>
    </div>
    
  </div>
  <div class="activity">
    <div class="content">
      <h2>دورة Canva</h2>
      <p>طفلك مبدع؟ عنده خيال واسع؟</p>
      جاهز يتعلّم كيف يحوّل أفكاره لتصاميم رهيبة؟
    </div>
  </div>
  <div class="activity">
    <div class="content">
      <h2>Canva انضموا معنا إلى كورس</h2>
      <p>مخصص للأطفال من عمر 8 إلى 15 سنة</p>
      بدون أي خبرة سابقة
      بأسلوب ممتع، تفاعلي، وبسيط
    </div>
  </div>
  <div class="activity">
    <div class="content">
      <h2>بإشراف المهندسة عبير حسن</h2>
      يتضمن الكورس:
      مبادئ التصميم الجرافيكي للأطفال
    </div>
  </div>

```

استخدام الألوان والخطوط بشكل احترافي

إنشاء تصاميم وبوسترات من الصفر

تعديل القوالب الجاهزة باحتراف

تصميم فيديوهات قصيرة وعروض تقديمية

مشاريع تطبيقية ممتعة وتحفيزية

! خلال أيام، رح يتعلموا يصمّموا بوسترات، بطاقات، عروض، وحتى فيديوهات
! الرهيبة... وبإبداعهم الخاص Canva كل هذا باستخدام منصة

...خلّي طفلك يدخل عالم التصميم بخطواته الأولى

. وخلّي الإبداع يحكي

</p> </للتواصل والاستفسار الاتصال على الرقم : 0934433913 عرض أقل

</div>

</div>

<div class="activity">

<div class="content">

<h2>دورة الروبوت</h2>

هل طفلك يحب الابتكار ويهوى استكشاف كيف تعمل الأشياء من حوله؟<p>

للأطفال من عمر 8 إلى Robotics LEGO Mindstorm EV3 الآن حان الوقت ليخطو أولى خطواته في عالم الروبوتات والبرمجة مع كورس . 16 سنة، بأسلوب عملي وتفاعلي يُحفّز الإبداع وينمّي مهارات التفكير المنطقي مع المهندسة عبير حسن

</p>

</div>

</div>

<div class="activity">

<div class="content">

<h2>دورة الروبوت</h2>

هل طفلك يحب الابتكار ويهوى استكشاف كيف تعمل الأشياء من حوله؟<p>

للأطفال من عمر 8 إلى Robotics LEGO Mindstorm EV3 الآن حان الوقت ليخطو أولى خطواته في عالم الروبوتات والبرمجة مع كورس . 16 سنة، بأسلوب عملي وتفاعلي يُحفّز الإبداع وينمّي مهارات التفكير المنطقي مع المهندسة عبير حسن

</p>

</div>

</div>

</div>

</div>

أما إعدادات CSS الخاصة بهذا القسم فهي:

.activities{

```

    margin-bottom: 10vh;
    display: flex;
    flex-direction: column;
    align-items: center;
}
.activities h3{
    color:var(--dark);
    font-size: 2rem;
    margin-bottom: 1rem;
    text-shadow: 0 0 5px var(--green);
}
.activities h6{
    color:var(--dark);
    font-size: 1rem;
    margin-bottom: 3rem;
}
.activity{
    background-color: var(--dark-gray);
    display: flex;
    border-radius: 3rem;
    overflow: hidden;
    box-shadow: 10px 10px 10px var(--dark);
    margin: 2rem;
    height: 350px;
}
.activity .content{
    padding: 1rem;
    flex:1;
    overflow-y: auto
}
.activity .content h2{
    color:var(--white);
    margin-bottom: 2rem;
}
.activity .content p{
    color:var(--gray);
    font-size: 1.2rem;
}
.activity img{
    flex:1;
    max-width: 50%;
}

```

يتألف القسم footer من عنصر **h3** وعنصر **div** يحتوي على أربعة أيقونات من مكتبة awesome.

```
<footer>
  <h3>تواصل مع معهد البيرم</h3>
  <div class="icons">
    <a href="#"><i class="fa-brands fa-facebook"></i></a>
    <a href="#"><i class="fa-brands fa-instagram"></i></a>
    <a href="#"><i class="fa-brands fa-twitter"></i></a>
    <a href="#"><i class="fa-brands fa-telegram"></i></a>
  </div>
</footer>
```

أما إعدادات CSS الخاصة بهذا القسم فهي:

```
footer{
  background-color:var(--dark-gray);
  min-height: 10vh;
  display: flex;
  align-items: center;
  justify-content: space-around;
  position: fixed;
  bottom: 0;
  width: 100%;
}
footer h3{
  color:var(--white);
  font-size: 1rem;

  text-align: center;
}
footer .icons{
  display: flex;
}
footer i{
  color:var(--gray);
  font-size: 1.5rem;
}
footer i:hover{
  color: var(--light-green);
}
```

6- إعدادات الاستجابة

إذا دخلنا إلى أدوات المطور باستخدام F12 ثم دخلنا إلى فاحص الاستجابة ثم بدأنا بالاختبار سنجد أن وضع العنوان h1 يحتاج إلى تعديل عندما يبدأ العرض بالانخفاض دون 900px:

```
@media screen and (max-width:900px){
```

```
.hero h1{
    width:80%;
}
}
```

قمنا بتغيير العرض إلى 80% بعد أن كان 40%.

عند الانخفاض دون الـ 600px يلاحظ ظهور عدة مشاكل بحاجة لمعالجة:

-الخط يصبح كبير نسبيا

-وضع h1 غير لائق

-الفعاليات تحتاج إلى فصل تغيير طريقة الظهور لتصبح الصورة تحت المحتوى من أجل كل فعالية.

بالنسبة للمشكلة الأولى والثانية الحل بسيط، أما بالنسبة للمشكلة الثالثة فنحن بحاجة إلى إلغاء القائمة nav-items، واستبدالها بزر عند الضغط عليه تظهر القائمة بشكل عمودي.

إذا أولا ينبغي علينا إضافة الزر البديل وإخفائه، لأجل ذلك نضيف الزر في الموقع المبين:

```
<nav>
    <button class="cta dark">تسجيل دخول</button>

    <ul class="nav-items">
        <li class="nav-item">
            <a href="#" class="nav-link">الصفحة الرئيسية</a>
        </li>
        <li class="nav-item">
            <a href="#" class="nav-link">من نحن</a>
        </li>
        <li class="nav-item">
            <a href="#" class="nav-link">فعالياتنا</a>
        </li>
    </ul>
    <i id="mobile-menu" class="fa-solid fa-bars"></i>
    
</nav>
```

ثم نقوم بإخفائه مع إعطائه بعض الخصائص عن طريق قاعدة CSS التالية:

```
nav #mobile-menu{
    color:var(--dark);
```

```
font-size: 2rem;
display: none;
cursor: pointer;
}
```

الان ولإستكمال اعدادات الاستجابة نضيف القاعدة التالية:

```
@media screen and (max-width:600px){
  html{
    font-size:90%;
  }
  .hero h1{
    width:90%;
  }

  .activity{
    flex-direction: column;
    height: fit-content;
  }
  .activity .content{
    max-height: 350px;
  }
  .activity img{
    max-width: 100%;
    max-height: 350px;
  }

  nav #mobile-menu{
    display:block;
  }
  nav .nav-items{
    display: none;
  }
  nav .nav-items .nav-item{
    padding: 1rem 0;
    font-size: 1.5rem;
  }
}
```

إذا اختبرنا استجابة الموقع الان سنراها مناسبة لكافة مقاسات الشاشات بشكل تقريبي طبعاً لأنه دائماً يوجد ما هو أفضل.

ما هي الخطوة التالية؟

الخطوة التالية هي معاودة ظهور القائمة عمودياً عند النقر على الزر.

إذا لدى إخفاء القائمة سنغير طريقة ظهورها عن طريق خاصية flex-direction لتصبح قاعدة css الأخيرة على النحو التالي:

```
@media screen and (max-width:600px){
  html{
    font-size:90%;
  }
  .hero h1{
    width:90%;
  }

  .activity{
    flex-direction: column;
    height: fit-content;
  }
  .activity .content{
    max-height: 350px;
  }
  .activity img{
    max-width: 100%;
    max-height: 350px;
  }

  nav #mobile-menu{
    display:block;
  }
  nav .nav-items{
    display: none;
    flex-direction: column;
    position: absolute;
    background-color: var(--green);
    width:100%;
    height: 35%;
    top:12vh;
    left:0;
    align-items: center;
  }
  nav .nav-items .nav-item{
    padding: 1rem 0;
    font-size: 1.5rem;
  }
}
```

الآن صحيح أن إعدادات ظهورها تغيرت لتلائم الحالة، إلا أننا لا ننسى أنها مخفية، فما الذي سيغير طريقة ظهورها من none إلى flex؟

هنا يأتي دور Java Script ..

لإنجاز ذلك نقول أن هذه القائمة عندما تكون من الصف active ستكون ظاهرة، وإلا فهي مخفية.

وهنا دور JS بالتحديد التي تمتلك الطرائق اللازمة لإضافة صنف لعنصر html أو إلغاء صنف أو إضافة/إلغاء خاصية أو ما شابه ذلك من مهام.

استجابة لهذا المطلب تصبح قاعدة CSS الأخيرة بالشكل التالي:

```
@media screen and (max-width:600px){
  html{
    font-size:90%;
  }
  .hero h1{
    width:90%;
  }

  .activity{
    flex-direction: column;
    height: fit-content;
  }
  .activity .content{
    max-height: 350px;
  }
  .activity img{
    max-width: 100%;
    max-height: 350px;
  }

  nav #mobile-menu{
    display:block;
  }
  nav .nav-items{
    display: none;
    flex-direction: column;
    position: absolute;
    background-color: var(--green);
    width:100%;
    height: 35%;
    top:12vh;
    left:0;
    align-items: center;
  }
  nav .nav-items .nav-item{
    padding: 1rem 0;
    font-size: 1.5rem;
  }
}
```

```
    }  
    nav .nav-items.active{  
        display: flex;  
    }  
}
```

نفتح الان ملف script.js جديد ونضع فيه المحتوى التالي:

```
const mobileButton=document.querySelector("#mobile-menu");  
const navItems=document.querySelector(".nav-items");  
mobileButton.addEventListener('click',() => {  
    navItems.classList.toggle("active");  
})
```

وقبل إغلاق الوسم body مباشرة نضمن ملف js الجديد بالطريقة التالية:

```
<script src="script.js"></script>
```

الان سنلاحظ أنه عند الضغط على زر القائمة ستظهر وتختفي القائمة بشكل منسجم.

الان وبهدف إخفاء القائمة عند الضغط على أي عنصر من عناصرها نضيف إلى الملف script.js الكود التالي:

```
const navLinks=document.querySelectorAll(".nav-link");  
navLinks.forEach((navl)=>{  
    navl.addEventListener('click',()=>{  
        navItems.classList.toggle("active");  
    })  
})
```