

[لطفا قسمت اول را در اینجا مطالعه بفرمائید](#)

گام سوم: افزودن یک button

در این مرحله یک button را به صفحه html اضافه می‌کنیم. button زمانی فعال می‌شود که هیچ متنی در فیلد input موجود نباشد. زمانی که کاربر بر روی دکمه کلیک می‌کند نام Meysam Khoshbakht را در کادر قرمز رنگ می‌نویسد. تگ <button> را بصورت زیر در زیر فیلد input ایجاد کنید

```
...
<div>
  <div>
    <input type="text" id="inputName" maxlength="15">
  </div>
  <div>
    <button id="generateButton">Aye! Gimme a name!</button>
  </div>
</div>
...
```

در زیر دستور import و بصورت top-level متغیر زیر را تعریف کنید تا یک ButtonElement در داخل آن قرار دهیم.

```
import 'dart:html';
ButtonElement genButton;
```

توضیحات

- ButtonElement یکی از انواع المنت‌های DOM می‌باشد که در کتابخانه dart:html قرار دارد
- اگر متغیری مقداردهی نشده باشد بصورت پیش فرض با null مقداردهی می‌گردد
- به منظور مدیریت رویداد کلیک button کد زیر را به تابع main اضافه می‌کنیم

```
void main() {
  querySelector('#inputName').onInput.listen(updateBadge);
  genButton = querySelector('#generateButton');
  genButton.onClick.listen(generateBadge);
}
```

جهت تغییر محتوای کادر قرمز رنگ تابع top-level زیر را به piratebadge.dart اضافه می‌کنیم

```
...
void setBadgeName(String newName) {
  querySelector('#badgeName').text = newName;
}
```

جهت مدیریت رویداد کلیک button تابع زیر را بصورت top-level اضافه می‌کنیم

```
...
void generateBadge(Event e) {
  setBadgeName('Meysam Khoshbakht');
}
```

همانطور که در کدهای فوق مشاهده می‌کنید، با فشردن button تابع generateBadge فراخوانی می‌شود و این تابع نیز با فراخوانی تابع setBadgeName محتوای badge یا کادر قرمز رنگ را تغییر می‌دهد. همچنین می‌توانیم کد موجود در updateBadge مربوط به

رویداد input فیلد input را بصورت زیر تغییر دهیم

```
void updateBadge(Event e) {
  String inputName = (e.target as InputElement).value;
  setBadgeName(inputName);
}
```

جهت بررسی پر بودن فیلد input می‌توانیم از یک if-else بصورت زیر استفاده کنیم که با استفاده از توابع رشته ای پر بودن فیلد را بررسی می‌کند.

```
void updateBadge(Event e) {
  String inputName = (e.target as InputElement).value;
  setBadgeName(inputName);
  if (inputName.trim().isEmpty) {
    // To do: add some code here.
  } else {
    // To do: add some code here.
  }
}
```

توضیحات

- کلاس String شامل توابع و ویژگی‌های مفیدی برای کار با رشته‌ها می‌باشد. مثل trim که فواصل خالی ابتدا و انتهای رشته را حذف می‌کند و isEmpty که بررسی می‌کند رشته خالی است یا خیر.
- کلاس String در کتابخانه dart:core قرار دارد که بصورت خودکار در تمامی برنامه‌های دارت import می‌شود
- حال جهت مدیریت وضعیت فعال یا غیر فعال بودن button کد زیر را می‌نویسیم

```
void updateBadge(Event e) {
  String inputName = (e.target as InputElement).value;
  setBadgeName(inputName);
  if (inputName.trim().isEmpty) {
    genButton..disabled = false
    ..text = 'Aye! Gimme a name!';
  } else {
    genButton..disabled = true
    ..text = 'Arrr! Write yer name!';
  }
}
```

توضیحات

- عملگر cascade یا آبشاری (...), به شما اجازه می‌دهد تا چندین عملیات را بر روی اعضای یک شی انجام دهیم. اگر به کد دقت کرده باشید با یک بار ذکر نام متغیر genButton ویژگی‌های disabled و text را مقدار دهی نمودیم که موجب تسریع و کاهش حجم کد نویسی می‌گردد.
- همانند گام اول برنامه را اجرا کنید و نتیجه را مشاهده نمایید. با تایپ کردن در فیلد input و خالی کردن آن وضعیت button را بررسی کنید. همچنین با کلیک بر روی button نام درج شده در badge را مشاهده کنید.

Pirate badge

Aye! Gimme a name!

Arrr! Me name is

Meysam Khoshbakht

گام چهارم: ایجاد کلاس PirateName

در این مرحله فقط کد مربوط به فایل dart را تغییر میدهیم. ابتدا کلاس PirateName را ایجاد می‌کنیم. با ایجاد نمونه ای از این کلاس، یک نام بصورت تصادفی انتخاب می‌شود و یا نامی بصورت اختیاری از طریق سازنده انتخاب می‌گردد.

نخست کتابخانه dart:math را به ابتدای فایل dart اضافه کنید

```
import 'dart:html';  
import 'dart:math' show Random;
```

توضیحات

- با استفاده از کلمه کلیدی show، شما می‌توانید فقط کلاسها، توابع و یا ویژگی‌های مورد نیازتان را import کنید.

- کلاس Random یک عدد تصادفی را تولید می‌کند

در انتهای فایل کلاس زیر را تعریف کنید

```
...  
class PirateName {  
}
```

در داخل کلاس یک شی از کلاس Random ایجاد کنید

```
class PirateName {  
  static final Random indexGen = new Random();  
}
```

توضیحات

- با استفاده از static یک فیلد را در سطح کلاس تعریف می‌کنیم که بین تمامی نمونه‌های ایجاد شده از کلاس مشترک می‌باشد

- متغیرهای final فقط خواندنی می‌باشند و غیر قابل تغییر هستند.

- با استفاده از new می‌توانیم سازنده ای را فراخوانی نموده و نمونه ای را از کلاس ایجاد کنیم

دو فیلد دیگر از نوع String و با نام‌های firstName_ و appellation_ به کلاس اضافه می‌کنیم

```
class PirateName {
  static final Random indexGen = new Random();
  String _firstName;
  String _appellation;
}
```

متغیرهای خصوصی با (_) تعریف می‌شوند. Dart کلمه کلیدی private را ندارد.

دو لیست static به کلاس فوق اضافه می‌کنیم که شامل لیستی از name و appellation می‌باشد که می‌خواهیم آیتی را بصورت تصادفی از آنها انتخاب کنیم.

```
class PirateName {
  ...
  static final List names = [
    'Anne', 'Mary', 'Jack', 'Morgan', 'Roger',
    'Bill', 'Ragnar', 'Ed', 'John', 'Jane' ];
  static final List appellations = [
    'Jackal', 'King', 'Red', 'Stalwart', 'Axe',
    'Young', 'Brave', 'Eager', 'Wily', 'Zesty'];
}
```

کلاس List می‌تواند شامل مجموعه ای از آیتم‌ها می‌باشد که در Dart تعریف شده است.

سازنده ای را بصورت زیر به کلاس اضافه می‌کنیم

```
class PirateName {
  ...
  PirateName({String firstName, String appellation}) {
    if (firstName == null) {
      _firstName = names[indexGen.nextInt(names.length)];
    } else {
      _firstName = firstName;
    }
    if (appellation == null) {
      _appellation = appellations[indexGen.nextInt(appellations.length)];
    } else {
      _appellation = appellation;
    }
  }
}
```

توضیحات

- سازنده تابعی همانام کلاس می‌باشد

- پارامترهایی که در {} تعریف می‌شوند اختیاری و Named Parameter می‌باشند. Named Parameter ها پارامترهایی هستند که جهت مقداردهی به آنها در زمان فراخوانی، از نام آنها استفاده می‌شود.

- تابع nextInt() یک عدد صحیح تصادفی جدید را تولید می‌کند.

- جهت دسترسی به عناصر لیست از [] و شماره‌ی خانه‌ی لیست استفاده می‌کنیم.

- ویژگی length تعداد آیتم‌های موجود در لیست را بر می‌گرداند.

در این مرحله یک getter برای دسترسی به pirate name ایجاد می‌کنیم

```
class PirateName {
  ...
  String get pirateName =>
    _firstName.isEmpty ? '' : '$_firstName the $_appellation';
}
```

توضیحات

- Getterها متدهای خاصی جهت دسترسی به یک ویژگی به منظور خواندن مقدار آنها می‌باشند.

- عملگر سه گانه ?: دستور میانبر عبارت شرطی if-else می‌باشد

- \$ یک کاراکتر ویژه برای رشته‌های موجود در Dart می‌باشد و می‌تواند محتوای یک متغیر یا ویژگی را در رشته قرار دهد. در رشته 'firstName the \$_appellation_' محتوای دو ویژگی _firstName و _appellation در رشته قرار گرفته و نمایش می‌یابند.

- عبارت (expr <=;) یک دستور میانبر برای { return expr; } می‌باشد.

تابع setBadgeName را بصورت زیر تغییر دهید تا یک پارامتر از نوع کلاس PirateName را به عنوان پارامتر ورودی دریافت نموده و با استفاده از Getter مربوط به ویژگی pirateName، مقدار آن را در badge name نمایش دهد.

```
void setBadgeName(PirateName newName) {
  querySelector('#badgeName').text = newName.pirateName;
}
```

تابع updateBadge را بصورت زیر تغییر دهید تا یک نمونه از کلاس PirateName را با توجه به مقدار ورودی کاربر در فیلد input تولید نموده و تابع setBadgeName را فراخوانی نماید. همانطور که در کد مشاهده می‌کنید پارامتر ورودی اختیاری firstName در زمان فراخوانی با ذکر نام پارامتر قبل از مقدار ارسالی نوشته شده است. این همان قابلیت Named Parameter می‌باشد.

```
void updateBadge(Event e) {
  String inputName = (e.target as InputElement).value;

  setBadgeName(new PirateName(firstName: inputName));
  ...
}
```

تابع generateBadge را بصورت زیر تغییر دهید تا به جای نام ثابت Meysam Khoshbakht، از کلاس PirateName به منظور ایجاد نام استفاده کند. همانطور که در کد می‌بینید، سازنده‌ی بدون پارامتر کلاس PirateName فراخوانی شده است.

```
void generateBadge(Event e) {
  setBadgeName(new PirateName());
}
```

همانند گام سوم برنامه را اجرا کنید و نتیجه را مشاهده نمایید.

نظرات خوانندگان

نویسنده:

محمد 92

تاریخ:

۱۰:۲۲ ۱۳۹۳/۰۲/۰۳

سلام و ممنون از مطلب خوبتون، فقط امکانش هست لینک هایی برای بنچمارک دارت و جاوا اسکریپت بزارید تا ببینیم کدوم بهتر عمل می کنند و کدوم حجم کمتری برای دانلود نهایی دارند

نویسنده:

میثم خوشبخت

تاریخ:

۱۸:۱۵ ۱۳۹۳/۰۲/۰۳

[لینک بنچمارک Dart, dart2js و JavaScript](#) برای مقایسه Performance هریک از آنها که نشون میده Dart امتیاز بالاتری رو کسب کرده