

Intro to Flutter

Barbora Plašovská
Mobile dev lead GDSC EPITA



Workshop timeline

- Dart & Flutter overview
- Live demo
- Practical work



What is Flutter?

What is Flutter?

- open source framework by Google for building multi-platform applications from a single codebase



Why Flutter?

Why Flutter?

- compiles to ARM or Intel machine code as well as JavaScript
- build and iterate quickly with Hot Reload
- very flexible, adaptive design
- powered by Dart (a language optimized for fast apps on any platform)



Dart

A basic dart program

```
// Define a function.
```

```
void printInteger(int aNumber) {  
    print('The number is $aNumber.');// Print to console.  
}
```

```
// This is where the app starts executing.
```

```
void main() {  
    var number = 42; // Declare and initialize a variable.  
    printInteger(number); // Call a function.  
}
```


Variables

`var name = 'Bob';`

`Object name = 'Bob';`

`String name = 'Bob';`

Default value

Without null safety

```
int? lineCount;  
assert(lineCount == null);
```

With null safety

```
int lineCount = 0;
```

Late

```
late String description;
```

```
void main() {  
    description = 'Feijoada!';  
    print(description);  
}
```

Final and const

```
final name = 'Bob'; // Without a type annotation  
final String nickname = 'Bobby';
```

Builtin types

- Numbers (`int`, `double`)
- Strings (`String`)
- Booleans (`bool`)
- Lists (`List`, also known as *arrays*)
- Sets (`Set`)
- Maps (`Map`)
- Runes (`Runes`; often replaced by the `characters` API)
- Symbols (`Symbol`)
- The value `null` (`Null`)

Some other important types . . .

- `Object`: The superclass of all Dart classes except `Null`
- `Enum`: The superclass of all enums
- `Future` and `Stream`: Used in [asynchrony support](#)
- `Iterable`: Used in [for-in loops](#) and in synchronous [generator functions](#)
- `Never`: Indicates that an expression can never successfully finish evaluating
- `dynamic`: Indicates that you want to disable static checking
- `void`

Type test operators

```
(employee as Person).firstName = 'Bob';
```

```
if (employee is Person) {  
    // Type check  
    employee.firstName = 'Bob';  
}
```

Arrow syntax

```
bool hasEmpty = aListOfStrings.any((s) {  
    return s.isEmpty;  
});
```

```
bool hasEmpty = aListOfStrings.any((s) => s.isEmpty);
```


Cascades

```
var button = querySelector('#confirm');  
button?.text = 'Confirm';  
button?.classes.add('important');  
button?.onClick.listen((e) => window.alert('Confirmed!'));  
button?.scrollIntoView();
```

Cascades

```
querySelector('#confirm')  
  ?..text = 'Confirm'  
  ..classes.add('important')  
  ..onClick.listen((e) => window.alert('Confirmed!'))  
  ..scrollIntoView();
```

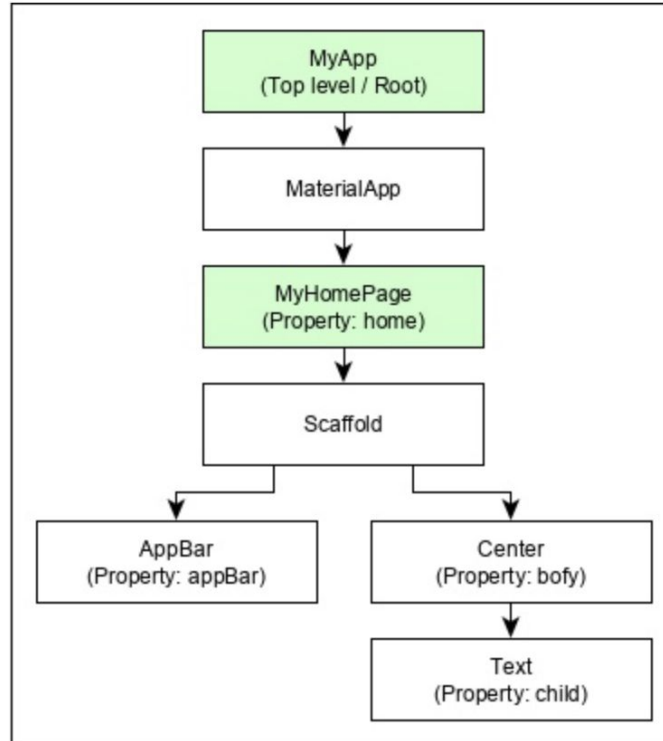
Going further

<https://dart.dev/guides/language/language-tour>

The background of the slide is a solid dark blue. In the top right corner, there is a decorative geometric pattern consisting of several squares and triangles in different shades of blue, including a lighter blue and a darker blue, creating a modern, abstract look.

Flutter

Everything. Is. a. widget.



Gestures & concept of state

- GestureDetector
- StatefulWidget
- StatelessWidget

```
class MyHomePage extends StatelessWidget {  
  MyHomePage({Key key, this.title}) : super(key: key);  
  final String title;  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(title: Text(this.title), ),  
      body: Center(child: Text('Hello World')),  
    );  
  }  
}
```

Going further

<https://docs.flutter.dev/>



Live demo

Let's start coding!





Kahoot!