Kotlin Workshop

Sondre Bakke
Konsulent

Bekk
B k
B

Hvem er vi?

Sondre
2 år i Bekk

Gaute
2 år i Bekk

Steffen
9 år i Bekk

Agenda

1. Lynkurs i Kotlin ⚡
2. Det progges 😎

Bekk
B k
B

Hva er Kotlin?

- Java, men kulere 😎
- Utviklet av JetBrains
- Java Interoperability

Bekk
B k
B

Lynkurs! ⚡

Bekk
B k
B

Funksjoner

```java
public int add(int a, int b) {
    return a + b;
}
```

```kotlin
fun add(a: Int, b: Int): Int {
    return a + b
}
```

```kotlin
fun add(a: Int, b: Int) = a + b
```

https://kotlinlang.org/docs/basic-syntax.html#functions

Funksjoner

```
public int add(int a, int b) {
    return a + b;
}
```

```
fun add(a: Int, b: Int): Int {
    return a + b
}

fun add(a: Int, b: Int) = a + b
```

https://kotlinlang.org/docs/basic-syntax.html#functions

Bekk

Funksjoner

```
public int add(int a, int b) {
    return a + b;
}
```

```
fun add(a: Int, b: Int): Int {
    return a + b
}

fun add(a: Int, b: Int) = a + b
```

https://kotlinlang.org/docs/basic-syntax.html#functions

Bekk
 B k
    B

Funksjoner som ikke returnerer noe

```
public void greet(String name, int age) {
    System.out.println("Hello " + name + ", you are " + age + " years old.");
}
```

```
fun greet(name: String, age: Int): Unit {
    println("Hello, $name! You are $age years old.")
}
```

```
fun greet(name: String, age: Int) = println("Hello, $name! You are $age years old.")
```

https://kotlinlang.org/docs/basic-syntax.html#functions

Bekk
 B k
    B

Funksjoner som ikke returnerer noe

```java
public void greet(String name, int age) {
    System.out.println("Hello " + name + ", you are " + age + " years old.");
}
```

```kotlin
fun greet(name: String, age: Int): Unit {
    println("Hello, $name! You are $age years old.")
}
```

```kotlin
fun greet(name: String, age: Int) = println("Hello, $name! You are $age years old.")
```

https://kotlinlang.org/docs/basic-syntax.html#functions

Bekk
 B k
   B

Funksjoner som ikke returnerer noe

```java
public void greet(String name, int age) {
    System.out.println("Hello " + name + ", you are " + age + " years old.");
}
```

```kotlin
fun greet(name: String, age: Int): Unit {
    println("Hello, $name! You are $age years old.")
}
```

```kotlin
fun greet(name: String, age: Int) = println("Hello, $name! You are $age years old.")
```

https://kotlinlang.org/docs/basic-syntax.html#functions

Variabler

```
val a: Int = 0
val b = 1

a = 1 // ikke lov

var iCanChange = 0
iCanChange = 1
iCanChange += 1
iCanChange++
```

https://kotlinlang.org/docs/basic-syntax.html#variables

Bekk
 B k
   B

Variabler

```kotlin
 val a: Int = 0
 val b = 1

 a = 1 // ikke lov

var iCanChange = 0
iCanChange = 1
iCanChange += 1
iCanChange++
```

https://kotlinlang.org/docs/basic-syntax.html#variables

Immutability

```kotlin
val numbers: MutableList<Int> = mutableListOf(1, 2, 3)
numbers.add(4) // [1, 2, 3, 4]
numbers.clear() // []
```

```kotlin
val numbers: List<Int> = listOf(1, 2, 3)
numbers.add(4) // ikke lov
```

Immutability

```kotlin
val numbers: MutableList<Int> = mutableListOf(1, 2, 3)
numbers.add(4) // [1, 2, 3, 4]
numbers.clear() // []


val numbers: List<Int> = listOf(1, 2, 3)
numbers.add(4) // ikke lov
```

Bekk
B k
B

Klasser

```kotlin
class BootcampCoach(val name: String, var yearsInBekk: Int) {
    fun introduce() {
        println("Hei, jeg heter $name og har vært i Bekk i $yearsInBekk år")
    }
}
```

```kotlin
val sondre = BootcampCoach("Sondre", 1)
println(sondre.name)
sondre.yearsInBekk = 2
sondre.introduce()
```

https://kotlinlang.org/docs/basic-syntax.html#creating-classes-and-instances

Bekk

Klasser

```kotlin
class BootcampCoach(val name: String, var yearsInBekk: Int) {
    fun introduce() {
        println("Hei, jeg heter $name og har vært i Bekk i $yearsInBekk år")
    }
}

val sondre = BootcampCoach("Sondre", 1)
println(sondre.name)
sondre.yearsInBekk = 2
sondre.introduce()
```

https://kotlinlang.org/docs/basic-syntax.html#creating-classes-and-instances

Data class

```kotlin
data class BootcampCoach(val name: String, var yearsInBekk: Int)
```

```
equals()
hashCode()
toString()
copy()
… og litt til!
```

https://kotlinlang.org/docs/data-classes.html

Bekk
 B k
    B

Argumenter

```
class Person(val name: String = "Anonym", val age: Int = 42)

Person("Sondre", 25)

Person("Sondre")

Person()

Person(age = 25)


fun greet(name: String = "world") = println("Hello, $name!")
```

https://kotlinlang.org/docs/idioms.html#default-values-for-function-parameters

Argumenter

```kotlin
class Person(val name: String = "Anonym", val age: Int = 42)

Person("Sondre", 25)

Person("Sondre")

Person()

Person(age = 25)


fun greet(name: String = "world") = println("Hello, $name!")
```

https://kotlinlang.org/docs/idioms.html#default-values-for-function-parameters

Bekk
B k
B

Argumenter

```kotlin
class Person(val name: String = "Anonym", val age: Int = 42)

Person("Sondre", 25)

Person("Sondre")

Person()

Person(age = 25)


fun greet(name: String = "world") = println("Hello, $name!")
```

https://kotlinlang.org/docs/idioms.html#default-values-for-function-parameters

If-expressions

```
fun maxOf(a: Int, b: Int): Int {
    if (a > b) {
        return a
    } else {
        return b
    }
}

fun maxOf(a: Int, b: Int) = if (a > b) a else b
```

https://kotlinlang.org/docs/control-flow.html#if-expression

Bekk
B k
B

Nullability

```java
int stringLength(String a) {
    return a.length();
}

void main() {
    stringLength(null); // Throws a `NullPointerException`
}
```

```kotlin
fun stringLength(a: String) = a.length

fun main() {
    stringLength(null) // ikke lov!
}
```

https://kotlinlang.org/docs/null-safety.html

Nullability

```java
int stringLength(String a) {
    return a.length();
}

void main() {
    stringLength(null); // Throws a `NullPointerException`
}
```

```kotlin
fun stringLength(a: String) = a.length

fun main() {
    stringLength(null) // ikke lov!
}
```

https://kotlinlang.org/docs/null-safety.html

Bekk
B k
B

Nullability

```
fun stringLength(a: String?): Int = if (a != null) a.length else 0

fun stringLengthOrNull(a: String?): Int? = a?.length

fun numberOrZero(a: Int?): Int = a ?: 0

fun stringLength(a: String?): Int = a?.length ?: 0
```

https://kotlinlang.org/docs/null-safety.html

Bekk
B k
B

Nullability

```kotlin
fun stringLength(a: String?): Int = if (a != null) a.length else 0

fun stringLengthOrNull(a: String?): Int? = a?.length

fun numberOrZero(a: Int?): Int = a ?: 0

fun stringLength(a: String?): Int = a?.length ?: 0
```

https://kotlinlang.org/docs/null-safety.html

Bekk
 B k
    B

Nullability

```kotlin
fun stringLength(a: String?): Int = if (a != null) a.length else 0

fun stringLengthOrNull(a: String?): Int? = a?.length

fun numberOrZero(a: Int?): Int = a ?: 0

fun stringLength(a: String?): Int = a?.length ?: 0
```

https://kotlinlang.org/docs/null-safety.html

Bekk
 B k
   B

Nullability

```kotlin
fun stringLength(a: String?): Int = if (a != null) a.length else 0

fun stringLengthOrNull(a: String?): Int? = a?.length

fun numberOrZero(a: Int?): Int = a ?: 0

fun stringLength(a: String?): Int = a?.length ?: 0
```

https://kotlinlang.org/docs/null-safety.html

Bekk
 B k
   B

Lambda-funksjoner

```
val coaches: List<BootcampCoach>

val oldCoaches = coaches.filter({ coach -> coach.yearsInBekk > 5 })
val oldCoaches = coaches.filter { coach -> coach.yearsInBekk > 5 }
val oldCoaches = coaches.filter { it.yearsInBekk > 5 }
```

https://kotlinlang.org/docs/lambdas.html

Bekk
B k
B

Lambda-funksjoner

```
val coaches: List<BootcampCoach>

val oldCoaches = coaches.filter({ coach -> coach.yearsInBekk > 5 })
val oldCoaches = coaches.filter { coach -> coach.yearsInBekk > 5 }
val oldCoaches = coaches.filter { it.yearsInBekk > 5 }
```

https://kotlinlang.org/docs/lambdas.html

Bekk
 B k
    B

Lambda-funksjoner

```
val coaches: List<BootcampCoach>

val oldCoaches = coaches.filter({ coach -> coach.yearsInBekk > 5 })
val oldCoaches = coaches.filter { coach -> coach.yearsInBekk > 5 }
val oldCoaches = coaches.filter { it.yearsInBekk > 5 }
```

https://kotlinlang.org/docs/lambdas.html

Extension functions

```kotlin
fun String.shout(): String {
    return "${this.uppercase()}!"
}

"hello".shout() // "HELLO!"
```

https://kotlinlang.org/docs/extensions.html

Bekk
 B k
   B

Scope-funksjoner: let, apply, run

```
class Rectangle(var width: Int, var height: Int) {
    var color: Color = Color.BLACK
    fun drawToScreen(): Unit = TODO("Ikke implementert")
}

var whiteSquare: Rectangle = Rectangle(10, 10).apply {
    color = Color.WHITE
}

var rect: Rectangle = Rectangle(10, 20).also { it.drawToScreen() }

var area: Int = Rectangle(10, 10).let { it.width * it.height }

var area: Int = Rectangle(10, 10).run { width * height }
```

https://kotlinlang.org/docs/scope-functions.html

Bekk
B k
B

Scope-funksjoner: let, apply, run

```
class Rectangle(var width: Int, var height: Int) {
    var color: Color = Color.BLACK
    fun drawToScreen(): Unit = TODO("Ikke implementert")
}


var whiteSquare: Rectangle = Rectangle(10, 10).apply {
    color = Color.WHITE
}
var rect: Rectangle = Rectangle(10, 20).also { it.drawToScreen() }

var area: Int = Rectangle(10, 10).let { it.width * it.height }

var area: Int = Rectangle(10, 10).run { width * height }
```

https://kotlinlang.org/docs/scope-functions.html

Bekk
B k
B

Scope-funksjoner: let, apply, run

```kotlin
class Rectangle(var width: Int, var height: Int) {
    var color: Color = Color.BLACK
    fun drawToScreen(): Unit = TODO("Ikke implementert")
}


var whiteSquare: Rectangle = Rectangle(10, 10).apply {
    color = Color.WHITE
}
var rect: Rectangle = Rectangle(10, 20).also { it.drawToScreen() }

var area: Int = Rectangle(10, 10).let { it.width * it.height }

var area: Int = Rectangle(10, 10).run { width * height }
```

https://kotlinlang.org/docs/scope-functions.html

Bekk
 B k
    B

Scope-funksjoner: let, apply, run

```kotlin
class Rectangle(var width: Int, var height: Int) {
    var color: Color = Color.BLACK
    fun drawToScreen(): Unit = TODO("Ikke implementert")
}

var whiteSquare: Rectangle = Rectangle(10, 10).apply {
    color = Color.WHITE
}
var rect: Rectangle = Rectangle(10, 20).also { it.drawToScreen() }

var area: Int = Rectangle(10, 10).let { it.width * it.height }

var area: Int = Rectangle(10, 10).run { width * height }
```

https://kotlinlang.org/docs/scope-functions.html

Scope-funksjoner: let, apply, run

```kotlin
class Rectangle(var width: Int, var height: Int) {
    var color: Color = Color.BLACK
    fun drawToScreen(): Unit = TODO("Ikke implementert")
}


var whiteSquare: Rectangle = Rectangle(10, 10).apply {
    color = Color.WHITE
}
var rect: Rectangle = Rectangle(10, 20).also { it.drawToScreen() }

var area: Int = Rectangle(10, 10).let { it.width * it.height }

var area: Int = Rectangle(10, 10).run { width * height }
```

https://kotlinlang.org/docs/scope-functions.html

Lynkurs overstått!

Spørsmål?

Tid for progging!

https://github.com/bekk/kotlin-workshop-bootcamp

1. Klon repoet
2. Åpne i IntelliJ
3. Åpne fila README.md
4. Følg instruksene der!

(Slides ligger i mappa `docs`)