

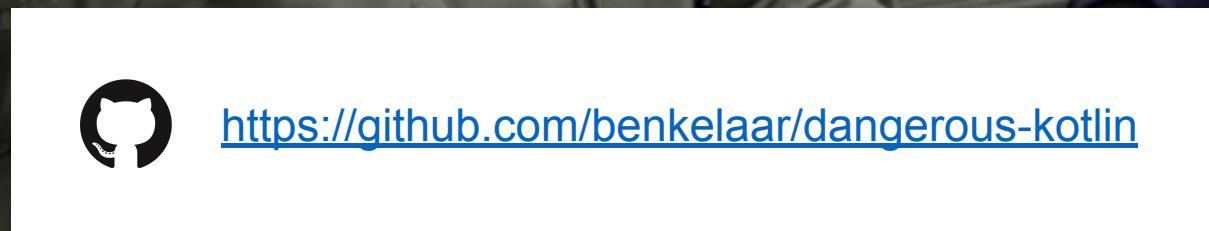
Kotlin
DEU DAY



THE DANGEROUS SIDE OF KOTLIN

BART ENKELAAR

KOTLINDEVDAY.COM





Ahold
Delhaize

bol.com





feestbeesten







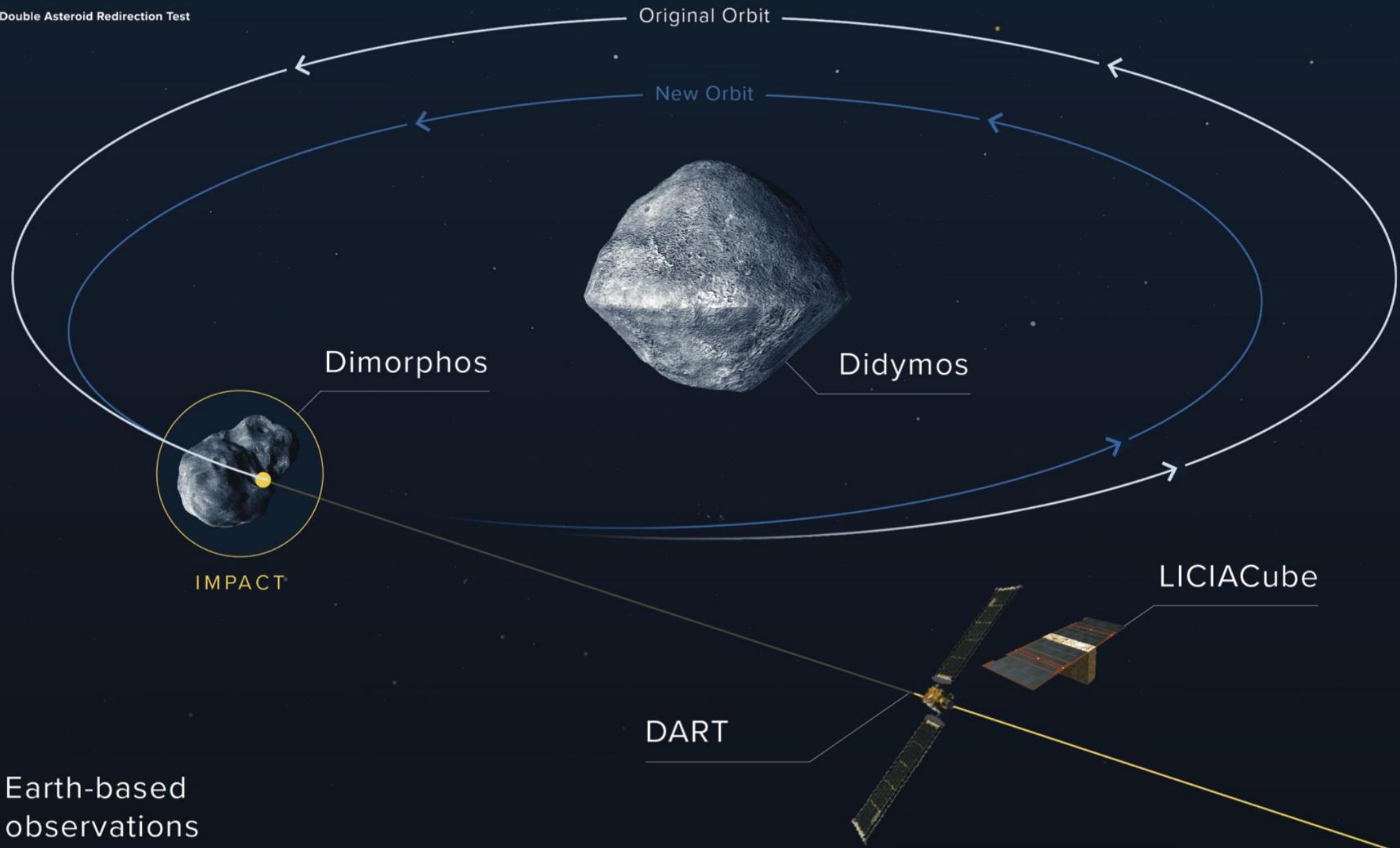




Double Asteroid Redirection Test



Earth-based
observations



Kotlin



DANGEROUS Kotlin

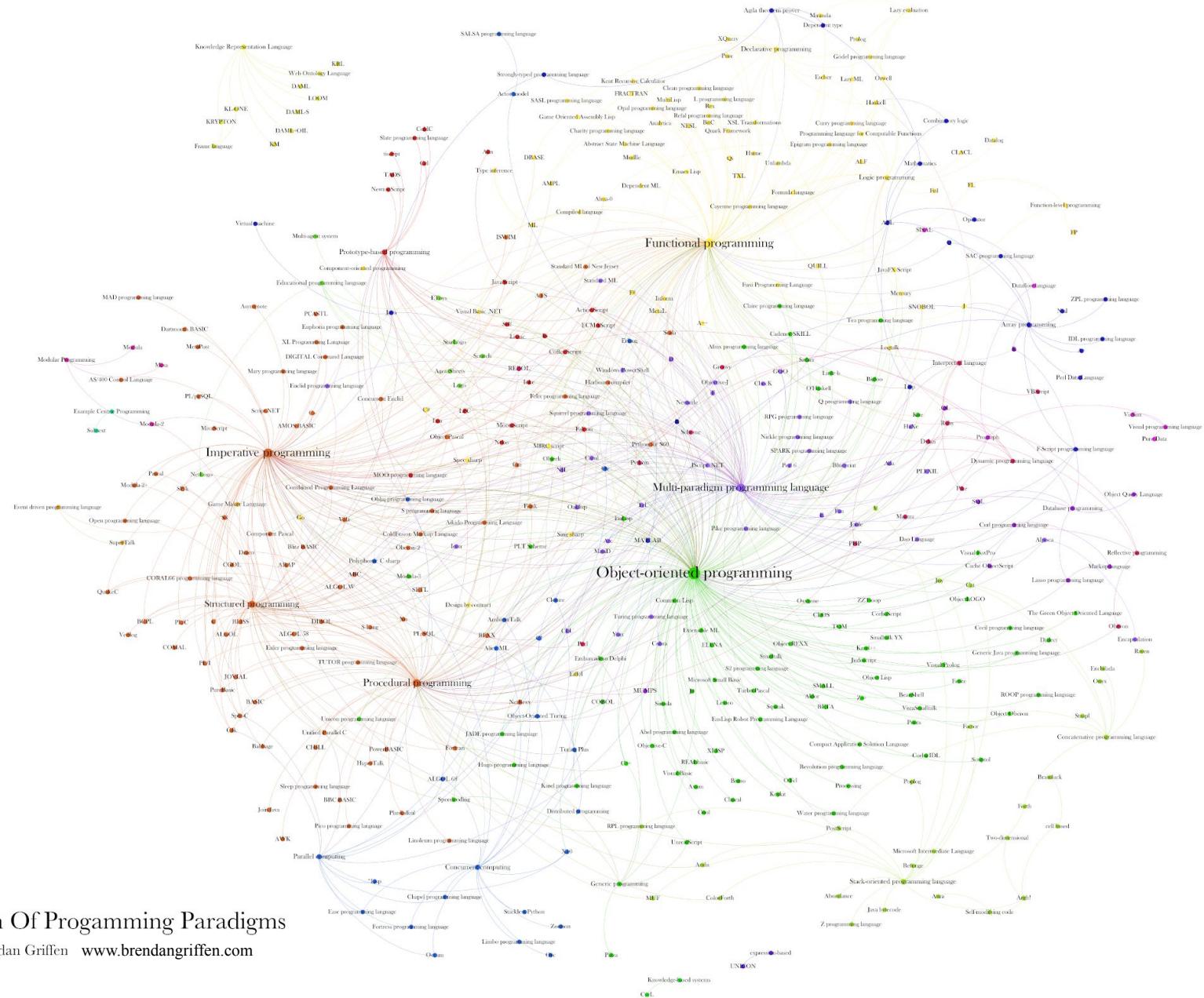






DANGEROUS Kotlin





The Graph Of Progarming Paradigms

By Brendan Griffen www.brendangriffen.com

- Fix our problem
- Easy to write
- Easy to read





I call it my
billion-dollar
mistake. It was
the invention of
the null reference
in 1965.

- Sir Tony Hoare, FRS, FREng

```
data class Coffee(val amount: Milliliters) {  
    fun merge(that: Coffee) = this + that  
  
    private operator fun plus(extra: Coffee) = Coffee(amount: amount + extra)  
}
```

```
typealias Milliliters = Double  
typealias CoffeeMerge = (Coffee, Coffee) -> Coffee
```

```
private val merge: CoffeeMerge = Coffee::merge
```

```
fun Double.toOunces() = this / 29.574
```

```
inline fun <reified T> T?.getClassString() = T::class.java.toString()
```



Extension functions

```
private fun String.asUrl() = URL(spec: this)
```

```
private fun String?.asUrl() = this?.let { URL(it) }
```

```
private fun String?.toUrl() =  
    try { this?.let { URL(it) } }  
    catch (e: MalformedURLException) { null }
```

```
data class PromotionReport(  
    val eans: Set<String>?,  
    val supplierId: String?,  
    val attribute: List<PromotedAttribute>?,  
    val promotedAttributesCount: Int?  
) : BeagleEntity
```

```
fun PromotionReport.getEans(): Set<String> = this.eans.orEmpty()
```

```
fun PromotionReport.getTraceId(): String? {  
    return this.attribute?.firstOrNull { it.id == AttributeId.TRACE_ID.value  
}
```

```
fun PromotionReport.getTraceIdOrThrow() = this.getTraceId() ?: throw Illega
```

```
val value = BigDecimal(val: "3.5")
```

```
val moreValue = value * 4
```

```
operator fun BigDecimal.times(int: Int) = this * BigDecimal(in
```

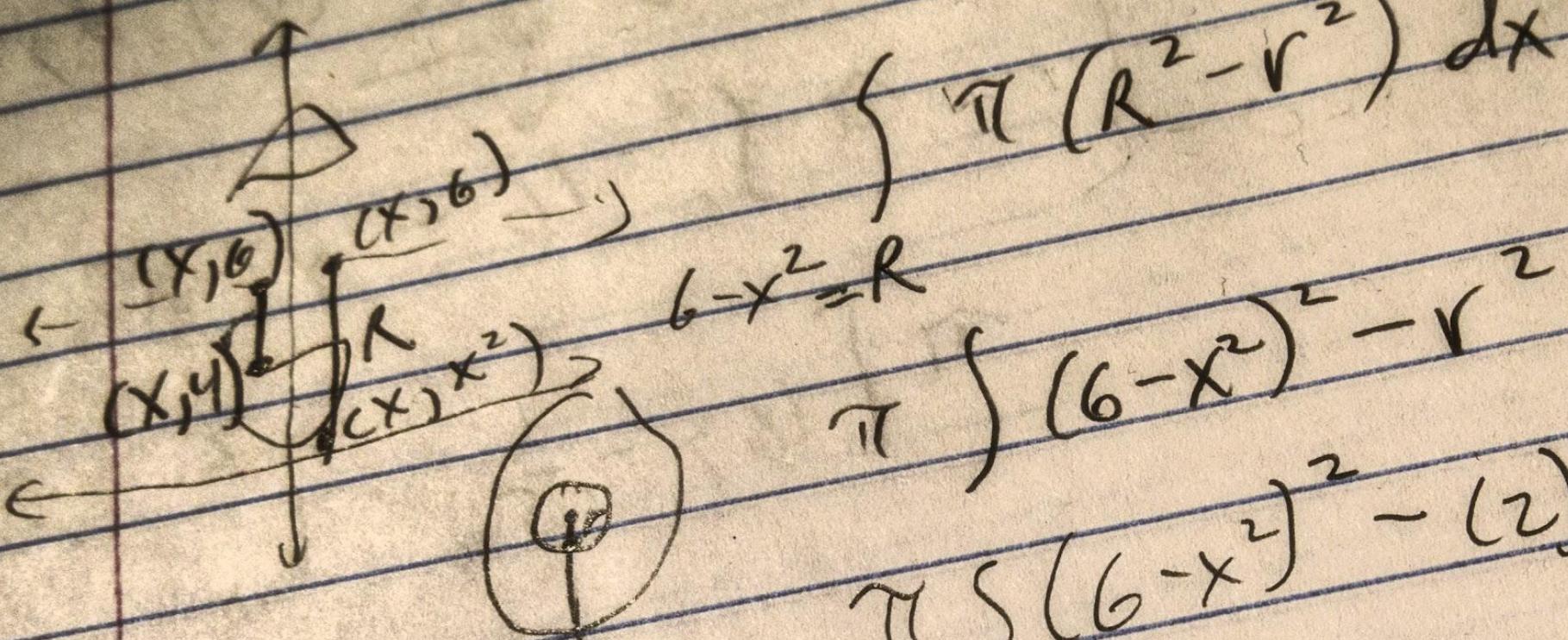
```
inline class Money(private val value: BigDecimal) {  
    constructor(value: Number) : this(BigDecimal(value.toString()))  
  
    fun toCurrency(type: CurrencyType) = Currency(money: this, type)  
  
    operator fun times(number: Number) = this * Money(number)  
    operator fun times(money: Money) = this * money.value  
    operator fun times(decimal: BigDecimal) = Money(value: value * decimal)  
}
```

Great DANGER

- Can introduce duplication
- Can break encapsulation
- Can interfere with OO principles

Operator overloading

```
inline class Money(private val value: BigDecimal) {  
    constructor(value: Number) : this(BigDecimal(value.toString()))  
  
    fun toCurrency(type: CurrencyType) = Currency(money: this, type)  
  
    operator fun times(number: Number) = this * Money(number)  
    operator fun times(money: Money) = this * money.value  
    operator fun times(decimal: BigDecimal) = Money(value: value * decimal)  
}
```



$$\pi \int_{(x,y)}^{(x+dx,y)} (6-x^2) \, dx$$

$$\pi \int_{(x,y)}^{(x+dx,y)} (6-x^2)^2 - r^2 \, dx$$

$$\pi \int_{(x,y)}^{(x+dx,y)} (6-x^2)^2 - (2)^2 \, dx$$

$$\pi \int_{-2}^2 (6-x^2)^2 - 4 \, dx$$

$$\pi \int_{-2}^2 36 - 12x^2 +$$

$$6-4$$

$$2=r$$

limits

$$x=4$$

$$4$$

$$x=x^2$$





• 100 euros

**StockTransactions
PriceTransactions**

StockValueTransactions

$$x\Delta S * y\Delta P = xy\Delta V$$

Great DANGER

- Only works if concept aligns
- Can surprise readers
- Risky for its seductive elegance

Lambdas
with
receiver

```
private fun configureTopicAndSubscription(topic: String, subscriptions: List<String>) {  
    pubSubAdmin.ignoringNotFound { deleteTopic(topic) }  
}  
  
}
```

*Whenever I read JavaScript I want to shout
“this is shit!”, but I can never figure out what
“this” refers to...*

- A tweet I once read who's author I've regrettably forgotten

```
private  
    a  
}  
} catch (e: NotFoundException) {  
    // ignore  
}
```

- 
- Great DANGER
- Obfuscates context scope
 - Doesn't pass “Smart code” test
 - Turns Kotlin into JavaScript





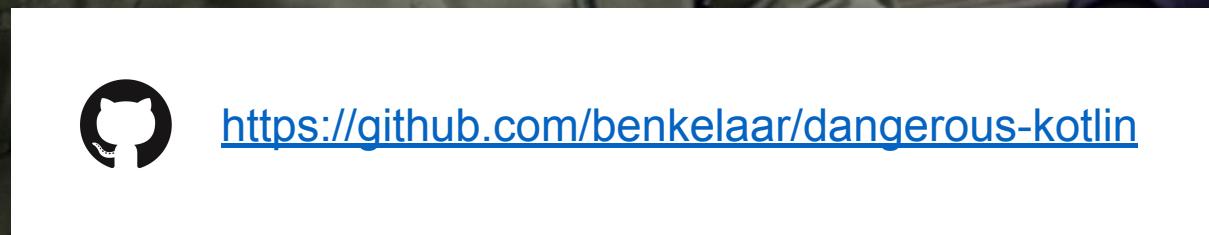


Thanks

Attributions

- “Dangerous pirate” by [Felix Lichtenfeld](#) via [Pixabay](#)
- “Danger stamp” by [Gerd Altmann](#) via [Pixabay](#)
- Math picture by Robert Couse-Baker from [Pxhere](#)
- Graph of programming paradigms by [Brendan Griffen](#)

- All other images Creative Commons license





Friendly Tech Chats

Bonus

