



# Modul I. Lesson 8

## Function I

Nasibali Abdiyev. Flutter Development

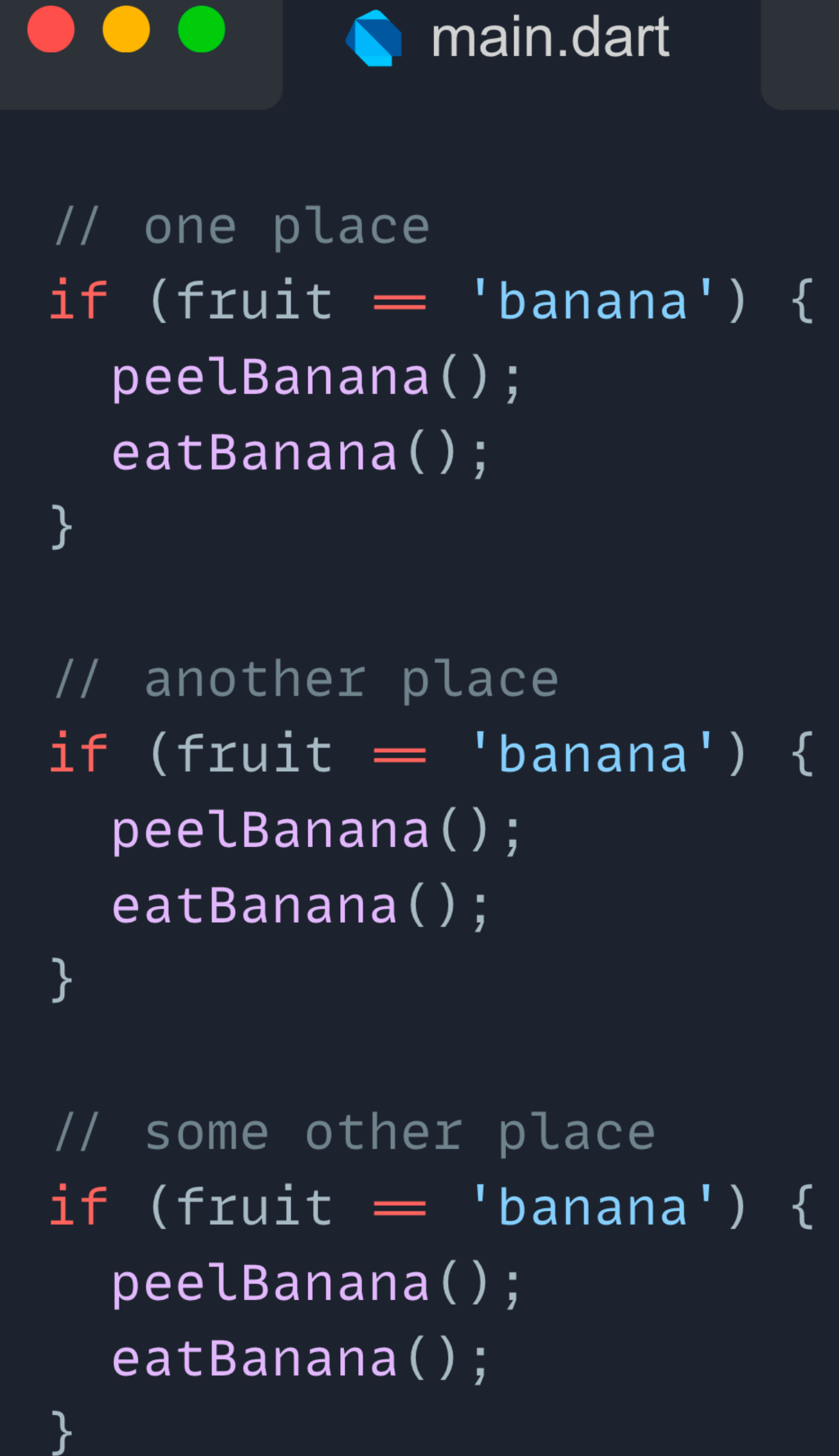
# Repeat the previous lesson

- Nested loop
- Break
- Continue
- Return
- Label

# Plan

- Function
- Why we use function
- Function return type

- Faraz qilaylik, sizda dastur davomida bir nechta joylarda takrorlangan kichik, foydali kod parchasi bor.
- Dasturlashda siz ko'pincha bir xil vazifani bir necha marta bajarishingiz kerak bo'ladi. Va siz dasturning hamma joylarida bir xil kodni nusxalash va joylashtirishni xohlamaysiz.
- Buni amalga oshirish uchun siz kodni funksiyaga aylantirasiz va xuddi shu kodni nusxalash(copy)/joylashtirish(paste) o'rniga undan foydalanasiz.



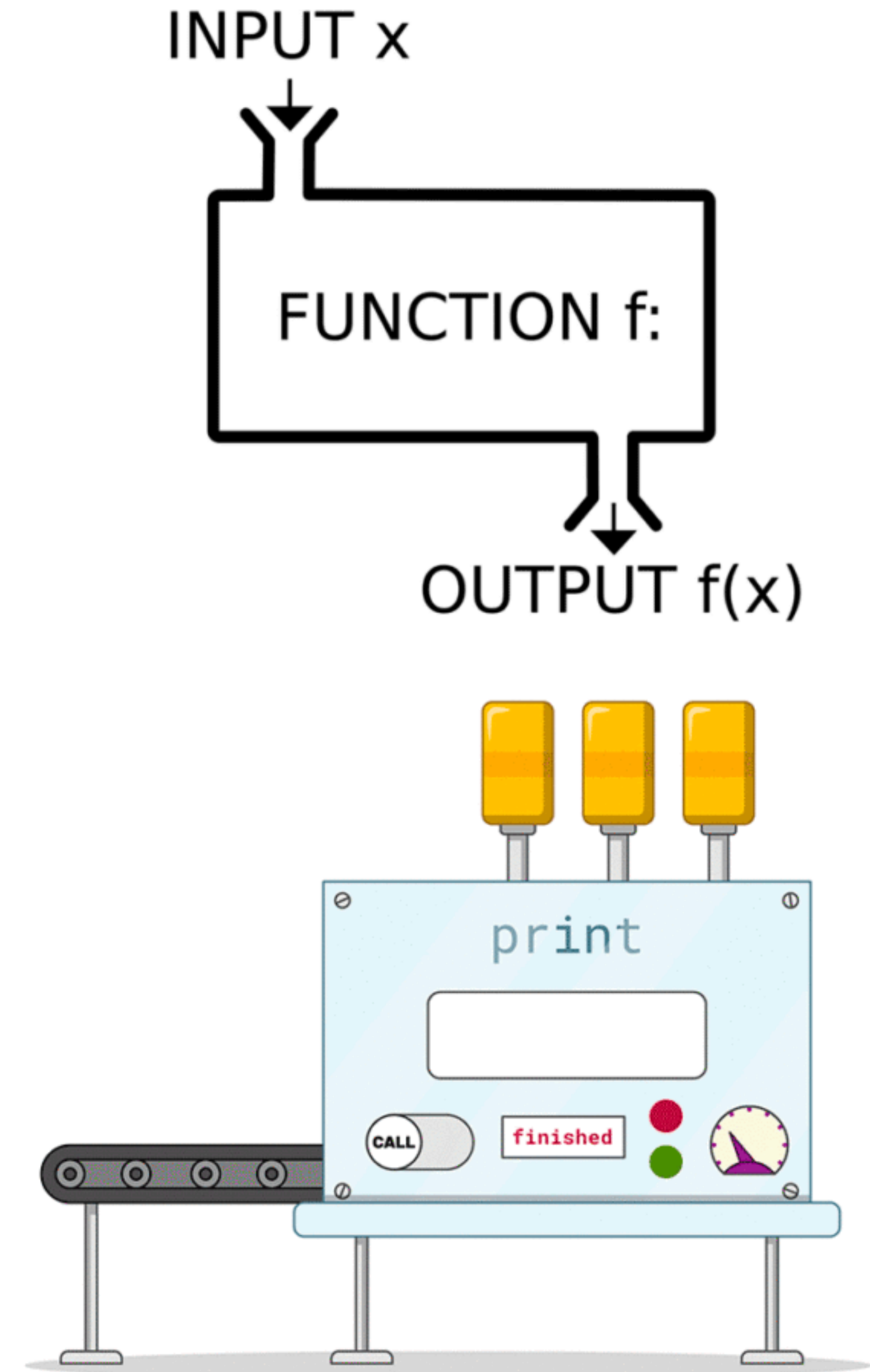
```
// one place
if (fruit == 'banana') {
  peelBanana();
  eatBanana();
}

// another place
if (fruit == 'banana') {
  peelBanana();
  eatBanana();
}

// some other place
if (fruit == 'banana') {
  peelBanana();
  eatBanana();
}
```

# Function

- Funksiya - ma'lum bir vazifani bajaruvchi, qandaydir nomga ega, bir yoki bir nechta qiymatni qabul qiluvchi, ishni tugatganidan keyin esa asosiy dasturga natija qiymatini qaytaruvchi qism dastur.
- Qisqasi: Funksiya ma'lum bir vazifani bajaradigan kod blokidir.



- Funktsiyalar dasturchilarga muammoni har biri ma'lum bir vazifani bajaradigan kichikroq bo'laklarga ajratish imkonini beradi.
- Funktsiya yaratilgandan so'ng, uning qanday ishlashi haqidagi tafsilotlar deyarli unutilishi mumkin. Shu tarzda tafsilot mavhumlashtiriladi(abstrakt).
- Funktsiya yaratilgandan keyin uni qayta-qayta ishlatish mumkin.
- Muayyan funktsiyadan foydalanish uchun siz quyidagilarni bilishingiz kerak: funksiya nomi, funksiya nima qiladi, funktsiyaga qanday argumentlar berish kerak va funksiya qanday natijani qaytaradi.
- Ammo e'tibor bering: agar siz shunchaki dasturingizda funktsiyadan foydalanmoqchi bo'lsangiz, uning ichida qanday ishlashini bilishingiz shart emas!

- Xuddi shunday, telefon bilan, qo'ng'iroq qilish uchun telefon tizimi haqida hamma narsani tushunishingiz shart emas.
- Funktsiyaning ichida qanday ishlashini bilishingiz kerak bo'lgan yagona vaqt - bu funktsiyaning yozishingiz yoki uning ishlashini o'zgartirishingiz kerak bo'lganda vaqtdir.
- Dasturni mavhum(abstrakt), qayta foydalanish mumkin bo'lgan qismlarga bo'lish qobiliyati aslida to'g'ri ishlaydigan katta dasturlarni yozish imkonini beradi.



# Functions

- Funksiyaning ikki turi mavjud:
  - ***Standard Library Function***: dartda oldindan belgilangan
  - ***User-defined Function***: foydalanuvchi tomonidan yaratilgan
- Biz asosan foydalanuvchi tomonidan hosil qilingan funktsiyalarga e'tibor qaratamiz.
- Foydalanuvchi tomonidan yaratilgan funksiya ma'lum bir vazifani bajarish uchun kodlarni guruhlarga ajratadi va bu kod guruhiga nom (identifikator) beriladi.
- Funktsiya dasturning istalgan qismidan chaqirilganda, uning barchasi funksiya tanasida aniqlangan kodlarni bajaradi.



# Syntax



main.dart

```
void main() {  
    // use:  
    functionName(arguments);  
}  
  
// declare:  
returnType functionName(parameters) {  
    // function body  
}
```

# Example



main.dart

```
void main() {  
    // calling a function  
    displayMsg("Hello everyone!");  
}  
  
// function declaration  
void displayMsg(String msg) {  
    print("Message: $msg");  
}
```

- `returnType` - bu funksiyada amal bajarilgandan so'ng natijani qaytaruvchi qiymatning turi hisoblanadi (misol uchun: `int`, `double`, `String...`). Agar funksiya hech qanday qiymat qaytarmasa uning hech qanday qiymat qaytarmasligini anglatish maqsadida `void` kalit so'zidan foydalaniladi.
- `functionName` - bu funksiyaning nomi bo'lib umumiy dasturda bajaradigan vazifasidan kelib chiqib foydalanuvchi tomondan unga qo'yilgan identifikator hisoblanadi. Va funksiya nomini yozishdan `lowerCamelCasedan` foydalaniladi.
- `parameters` - funksiyaga tashqaridan kirib keluvchi qiymatni o'zlashtiruvchi yani qabul qilib oluvchi o'zgaruvchi yoki o'zgarmaslar hisoblanadi. Parametrlarga ham nom berish oddiy o'zgaruvchi kabi `lowerCamelCase` orqali amalga oshiriladi. 0 tadan `n` tagacha bo'lishi mumkin.
- `arguments` - bu funktsiyaning chaqirayotganda va foydalanganimizda unga o'tadigan qiymat yoki o'zgaruvchi.

# Example



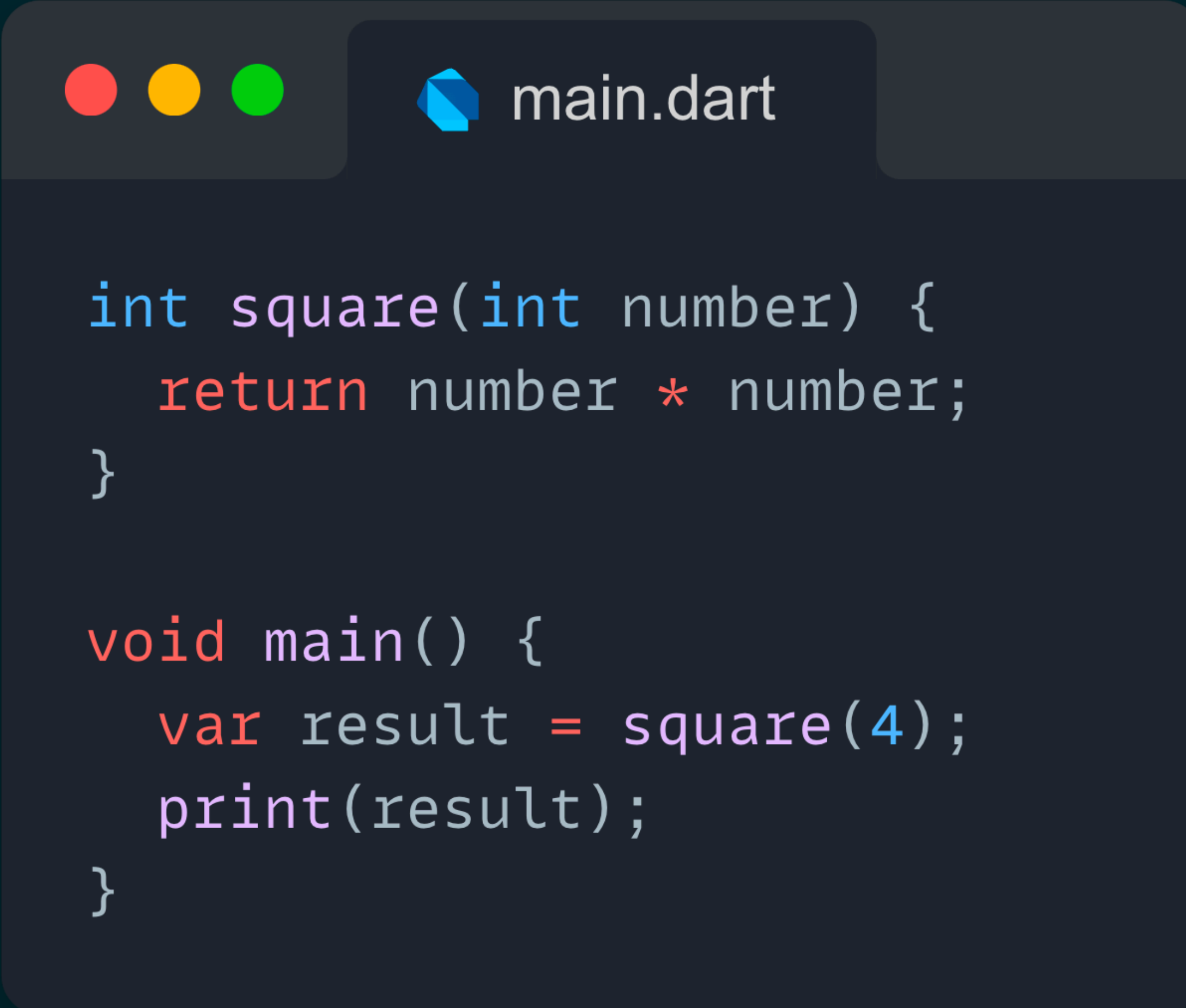
 main.dart

```
void main() {  
    int numberOne = 12;  
    int numberTwo = 13;  
  
    // calling a function  
    int result = add(numberOne, numberTwo);  
    print(result);  
}  
  
// function declaration  
int add(int a, int b) {  
    return a + b;  
}
```

# Some concepts related to function:

- ***Declare***: funktsiya deklaratsiyasi kompilyatorga funktsiya nomi, qaytish turi va parametrlari haqida xabar beradi.
- ***Define***: funktsiya ta'rifi muayyan vazifa nima va qanday bajarilishini belgilaydi. Funktsiyaning tanasini ta'minlaydi.
- ***Call***: funktsiyani bajarish yani foydalanish uchun uni chaqirish kerak.
- ***Return***: bajarilgan ishdan so'ng funktsiya natijani qaytarishi. Funktsiya faqat bitta qiymat qaytaradi
- ***Parameter***: parametrlar funktsiyalarga qiymatlarni uzatish mexanizmidir.
- Dartda ***function overloading***(qayta yuklanuvchi funktsiyalar) yo'q.

# Example



```
int square(int number) {  
    return number * number;  
}  
  
void main() {  
    var result = square(4);  
    print(result);  
}
```

# Exercise

1. Kiritilgan son juft yoki toqligini chiqaruvchi void funksiya tuzing va uni asosiy dasturda ishlatiing.
2.  $a$  va  $b$  son kiritiladi va bu yerda  $a \leq b$  bo'lib,  $a$  dan  $b$  gacha bo'lgan sonlar yig'indisini qaytaruvchi funksiya tuzing. Ex:  $a = 10, b = 15 \Rightarrow 10 + \dots + 15$



# Return Type

- Funksiyalar qiymat qaytarish yoki qaytarmasligiga qarab ikki turga bo'linadi:
  - void turli funksiyalar bo'lib hech qanday qiymat qaytarmaydi.
  - return turli funksiyalar bo'lib aniq bir turdagi qiymatni qaytaradi.
- Bazida funkسيyani ko'rinishiga qarab quyidagicha ajaratishadi(bu ixtiyoriy):
  - No Parameter And No Return Type
  - Parameter And No Return Type
  - No Parameter And Return Type
  - Parameter And Return Type



main.dart

// Type I.

```
void functionName( params ) {  
    function body  
}
```

// Type II.

```
dataType functionName( params ) {  
    function body  
    return data;  
}
```

functionName  $\Rightarrow$  camelCase and function name mean **this** what does **this** **do**

parametr  $\Rightarrow$  pass values to functions

function body  $\Rightarrow$  group code operation

**return** data  $\Rightarrow$  function result

# Example



main.dart

```
String currentDate() {  
    DateTime now = DateTime.now();  
  
    String convertedDateTime = "${now.year.toString()}"  
        "-${now.month.toString().padLeft(2, '0')}"  
        "-${now.day.toString().padLeft(2, '0')}"  
        " ${now.hour.toString().padLeft(2, '0')}"  
        " :${now.minute.toString().padLeft(2, '0')}";  
    return convertedDateTime;  
}
```

# Example



main.dart

```
bool isValidPassword(String password) {  
    return password.length ≥ 8;  
}
```

# Exercise

1. Berilgan 3 ta sonning o'rta arifmetigini hisoblab, natija sifatida qaytaruvchi funksiya tuzing.
2. Berilgan 3 ta sonning eng kattasini aniqlab, natija sifatida qaytaruvchi funksiya tuzing.

# Summary

# Interview Questions

1. Funksiya nima?
2. Funksiyadan foydalanish bizga qanday imkoniyat va qanday muammolarni hal qilishi imkonini beradi? (takror foydalanish, abstaktsiya, vazifalarni kichikroq bo'laklarga ajratish).
3. Umumiy olib qaraganda funksiyalar necha turga bo'linadi?
4. Standard Library Functions - bu qanday funksiya?
5. User-defined Function - bu qanday funkisya?
6. Dart tilida function overloading haqida ma'lumot bering.
7. Funksiyalar qiymat qaytarish yoki qaytarmasligiga ko'ra necha turga bo'lamiz?



8. Funksiyaning umumiy sintaksisida quyidagi elementlar nima vazifani bajaradi:

- return type
- function name
- parameters
- function body

9. Funksiya bilan bog'liq bazi tushunchalar nima ma'noni anglatadi:

- declare
- define
- call

10. return kalit so'zi qachon, qayerda va nima maqsadda ishlatiladi.

11. void kalit so'zi qachon, qayerda va nima maqsadda ishlatiladi.

12. parametr va argument nima? Ularning farqini tushuntiring.

# Homework

1. Foydalanauvchi tomonidan kiritilgan son tub yoki tub son emasligini aniqlovchi bool toifali funksiya tuzing. Agar parametrdagi kiritilgan son tub bo'lsa true, aks holda false qaytarsin.
2. Berilgan sonni teskarisini chiqaruvchi return type li funksiya tuzing.
3. Kiritilgan a va b sonni EKUB va EKUK ini hisoblovchi ikkita return type li funksiya tuzing.
4. Berilgan sonni faktorialini hisoblovchi return typeli funksiya tuzing.
5. N natural soni parametr sifatida qabul qilib. 1 dan N gacha bo'lgan tub sonlarni ekranga chiqaruvchi void typeli funksiya tuzing.

6. Berilgan butun sonning palindrome ekanligini aniqlaydigan funksiya tuzing.  
`isPalindrome( 121) → true, isPalindrome( 123) → false, isPalindrome ( -121) → false.`
7. RGB formatdagi ranglarni HEX formatga o'girib beruvchi funksiya tuzing: String  
`rgbToHex(int r, int g, int b) {/* your code */}`
8. Perfect Number. Agar berilgan son o'zining musbat bo'luvchilarining yig'indisiga teng bo'lsa bu son "Perfect Number" deyiladi. Berilgan son "Perfect Number" bo'lsa true bo'lmasa false qiymat chiqaradigan funksiya yarating.  
`checkPerfectNumber(28); // true => 28 = 1 + 2 + 4 + 7 + 14`
9. Haroratni Farengeytda darajada berganda, uni Selsiy darajaga o'giradigan funksiya yozing. (Bu yerda gap havo harorati haqida. Searching...)
10. x, y, z sonlari berilgan. Shu sonlar uchburchakning ichki burchaklari ekanligini aniqlaydigan funksiya tuzing. Agar berilgan sonlar uchburchakning ichki burchaklari bo'lsa true aks holda false qiymat qaytarsin.

11. Berilgan  $n$  sonigacha bo'lgan barcha kvadrat sonlarni ekranga chiqaruvchi method yarating. Masalan: `printSquares(30)` methodi ekranga 1, 4, 9, 16, 25 sonlarni ekranga chiqarsin.
12.  $a$  sonini natural  $n$  darajasini qaytaradigan `pow(a, n)` nomli funksiyatuzing. Matematik kutubxona funksiyalaridan foydalanilmasin.

# Q&A

**Thank you for your time!**