Timings

lunch break -- 2 pm -- 2:45 pm

1st tea break -- 4:45pm

2nd tea break

Flutter ---- React Native

Hybrid/ cross platform ------ Native apps

1. Dart (much easier to learn)
2. google
3. Hot Reload n Hot Restart
4. Material / widgets
5. performance and speed
6. single code based (high productivity / fast building app / learning curve is short)

dart 2013 -- 2015

2017 --

android
ios
iot
web
desktop

```
 String myName = "Ansari";
  print("my name is $myName");
  print("lenght is ${myName.length}");

  var age = 100;
  print("my age is $age");
```

```
String city ="" ;
late String state;

void main(){
  state = "Karnataka";
  print(state);
  String country;
  country = "india";
  print(country);
  print(city);
// null safety
  String? name = null;
  name = "ansari";
  print(name.length);

  int? age;
  int? newAge = age;
}
```

```
String city ="" ;
late String state;
var id;
var account = null;

void main(){
  print(account);
  state = "Karnataka";
  id = 101;
  print(id);
  print(state);
  String country;
  country = "india";
  print(country);
  print(city);
// null safety
  String? name = null;
  name = "ansari";
  print(name.length);

  int? age;
  int? newAge = age;
```

}

//final and constant
final age1 = 100; // mem alloc will done when u use it
const age2 = 100; // mem alloc is done at compile/declare

int a = 1, b = 2;
final ag = a;
print(ag);

const agc = b;

List

```
void main(){

 // Growable

 var mylist = [34,55,66,77,554];
 /*for(int i = 0; i<=mylist.length-1; i++){
   print(mylist[i]);
 }*/

 for(var i in mylist){
   print(i);
 }

 // fixed list
 var mlist = new List.filled(5, null, growable: false);

}
```

Sets

```
var myset = {34,55,66,77,554,77};
print("set is ${myset}");

 for(var i in mylist){
   print(i);
 }
```
—-----------------------------

```
var myMap = {"id":101, "name":"ansari",
"city":"blore"};
print(myMap);

print(myMap['id']);

myMap.forEach((key, value) {
 print("my key is $key and the value is $value");
});
```

```
/*void playing(){
  print("playing fun");
}

playing();*/

/*getDetails(String name, int age, String city){
  print("my name is $name  and age is $age and city is $city");
}

getDetails("ansari", 33, "blore");*/

//optional
/*  getDetails(String name, [int? age, String? city]){
    print("my name is $name  and age is $age and city is $city");
  }

  getDetails("ansari", 33);
  getDetails("ravi");*/

/*  getDetails(String name, {int? age = 100, String? city = "UnKNOWN"}){
    print("my name is $name  and age is $age and city is $city");
  }

  //getDetails("ansari", 33);
  getDetails("ravi");*/

// position
  getDetails(String name, {int? age,  String? city = "kar"}){
    print("my name is $name  and age is $age and city is $city");
  }

  //getDetails("ansari", 33);
  getDetails("ravi", age: 33);

----------------------





sum1(a,b){
  return a + b;
}

// fat arrow
sum(a,b) => a + b;

// lambda fun
var myMul = (a,b) => a * b;

void main() {

  var res = sum(4,5);
  print(res);

  ({int a = 11 , int b = 12}){
    print("addintion is ${a + b}");
```

```
 }();

 (){
  print("TEST");
 }();

}
```

—------------------------------

```
sum(a,b) => print(a + b);
// higher order fn
higherOderFunction(x,y, Function myfun){
 print("THIS IS higher order fun");
 myfun(x,y);
}

void main(){

 higherOderFunction(4, 5, sum);

}
```

—------------------------------

```
// returning fun as parameter
Function taskToPerform(){
 Function multiply = (int a, int b) => a * b;
 return multiply;
}
void main(){

var myFun =  taskToPerform();
    var res =  myFun(4,33);
    print(res);
}
```

—------------------------------

```
void main(){
  // closure
  var age = 22;
  print("main fun age is $age");

  void InnerFun(){
    //lexical scope
    age = 10;
    print("inner fun age is $age");
  }

}
```

```
typedef myFun(int a,  int b);

sum(a,b) => print(a + b);

void main(){

  myFun mf = sum;
  mf(1,2);
}
```

—---

```
class Tiger{
 var name = "TIGER";
 var age = 22;

  eating(){
   print("tiger eating");
  }
}

void main(){
Tiger tig = Tiger();
var tiger = Tiger();
Tiger().eating();
print(tig.name);
}
```

—---
```
class Tiger{

 Tiger(String city){
   print("tiger city is $city");
 }
 Tiger.customCon(int id, String country){
   print("tiger id is $id and country is
$country");
 }

 Tiger.customCon1(int num, String state){
   print("tiger id is $num and country is $state");
 }
 var name = "TIGER";
 var age = 22;

  eating(){
   print("tiger eating");
 }
}

void main(){

 var tiger = Tiger("blore");

 var tig = Tiger.customCon(101, "INDIA");

}
```
—---

Mixins

```
class Student{
 void name(){}
 void age(){}
 void roll(){}
}
class Teacher{
 void name(){}
 void age(){}
 void subject(){}
}

mixin Name{
 void name(){}
}

mixin Age{
 void age(){}
}

class StudentOne with Name, Age{
 void roll(){}
}
class TeacherOne with Name, Age{
 void subject(){}
}
```

—---

```
class Tiger{

 var city = "New York";

 var name = "TIGER";
 var age = 22;

//Meta
 @deprecated
 /** use new method playingNeating **/
  eating(){
   print("tiger eating");
 }

 playingNeating(){
    print("tiger is playing and eating");
 }
}

void main(){
var tiger = Tiger();
tiger.eating();


}
```

```
Future<String> getData(){
 return Future.delayed(Duration(seconds: 5), (){
   return "Data received from server";
 });
}

void main() async{
 print("main fun");
 print( await getData());

}
----------

Future<String> getData(){
 return Future.delayed(Duration(seconds: 5), (){
   return "Data received from server";
 });
}

void main() {
 print("main fun");
 getData().then((value){
   print(value);
 });
 print("other part of the program is executing");
---------------------------
```

```
Future<String> getData(){
 return Future.delayed(Duration(seconds: 5), (){
   throw "Server not responded";
 });
}

void main() {
 print("main fun");
 getData().then((value){
   print(value);
 });
 print("other part of the program is executing");

}
```
