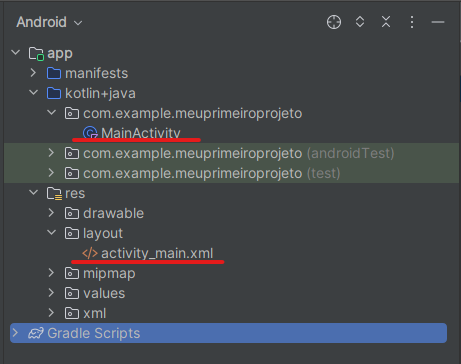
**CURSO DE KOTLIN E JAVA**

Ao iniciar um projeto, caso os arquivos principais não abram, é só acessar essas pastas:

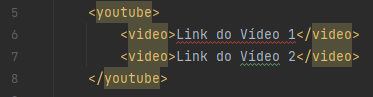


**Estrutura de um projeto**

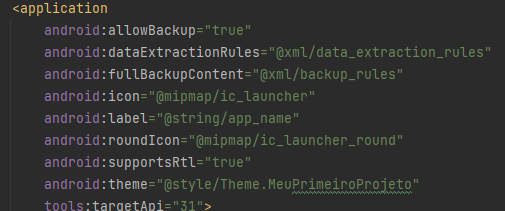
O Gradle Scripts é usado para fazer rotinas e adicionar recursos ao projeto.

O Android Manifest é usado para fazer configurações no aplicativo, já o active\_main é usado para criar a interface do app.

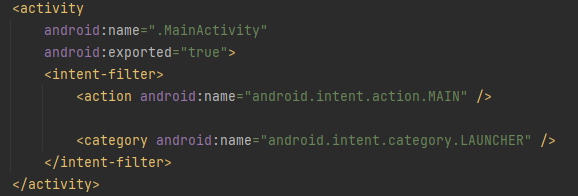
A abertura de uma tag é um símbolo de menor que, tag, símbolo de maior que.  
Ex:



Dentro do Android Manifest é possível pedir permissões para o usuário. Como por Exemplo: Utilizar a câmera ou até pedir a sua localização.

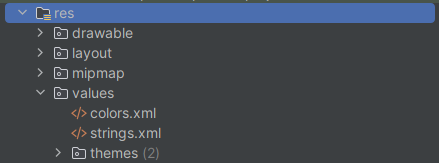


android:icon – É possível mudar qual ícone que irá aparecer no seu app.  
android:label – É o nome do seu app, ele será exibido na lista de todos os apps do smartphone.  
android:roundIcon – É o formato que seu app ficara. Simplificando seria as bordas do ícone.



O activity é usado para criar uma tela no seu projeto. Com determinadas configurações é possível definir qual será a tela de abertura do app.

**RES**



Essa pasta possui diversos arquivos interessantes, pois com ele é que podemos fazer a nossa personalização do app.

Drawable – Essa pasta é utilizada para alocar as imagens que será utilizada no app.  
Layout – É a pasta que armazena o activity\_main.xml  
Mipmap – É onde são armazenados os vetores que serão utilizados no app.  
Values -> Colors – São armazenadas as cores em formas de variáveis.  
Values -> Strings – São as variáveis, sendo elas textos que podem ser utilizados de uma forma mais simplificada para alterar um texto que está em diversos lugares.  
Themes – É uma formação de cor em várias regiões padrões do app.

**Layouts**

Os Layouts são sempre alinhados ao conto superior esquerdo, caso não sejam ligadas as extremidades da tela ou de elementos.

**Interoperabilidade**

O Java e o Kotlin podem trocar códigos entre si, pois eles têm interoperabilidade. É possível utilizar ações de um em outro, mas é preciso que os códigos sejam de acordo com a linguagem que está sendo utilizada.

**JAVA**

**Algoritmo**É uma sequência finita de ações executáveis que visam obter a solução para um tipo de problema.

**A linguagem Java é utilizada para diversas atividades**

* Programas de Computadores (Desktop e Web)
* Criação de Jogos
* Sistemas Automotivos
* Automação de Residências
* Máquinas Industriais
* TVs, Celulares, Tablets
* E muito mais

**Editores Online de Java**

<https://www.browxy.com/>

<https://www.jdoodle.com/online-java-compiler>

<https://replit.com/languages/java10>

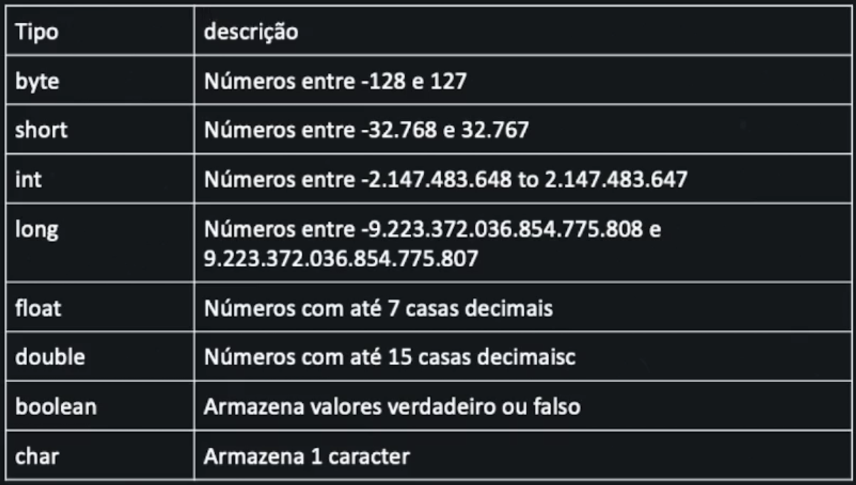
<https://www.w3schools.com/java/java_compiler.asp>

**Variáveis e Constantes**

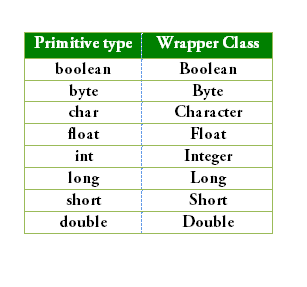
Como definir variável?  
String nomeDaVariavel = Resultado; - Variável String.  
Integer nomeDaVariavel = Resultado; - Variável Inteira.

Como definir constantes?  
final Double Pi = 3,14... – Para definir uma constante, é preciso colocar a palavra final e escolher o tipo. Constantes são normalmente declaradas com as letras maiúscula.

**Tipos de Variáveis (Tipos Primitivos)**



**Classes Wrapper**As Classes Wrapper é utilizada para funções especificas, mas o mais importante é priorizar os tipos primitivos.



**Operadores Aritméticos**

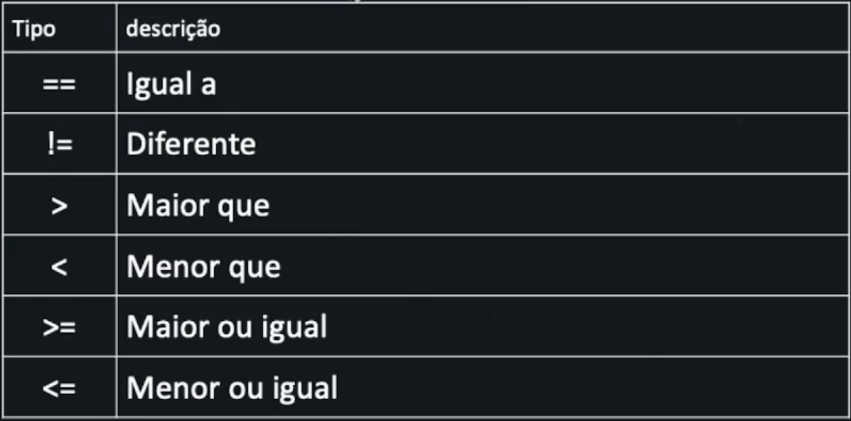
**Básicos**

****

**Ordem de Precedência**

1. **Parênteses**
2. **Multiplicação e Divisão**
3. **Subtração e Adição**

**Operadores relacionais**



**Operadores Lógicos**

****

**Estruturas condicionais (If)**

if(condição) { **SE**

}else if(condição) { SE SENÃO

}else { SENÃO

}

**Estruturas condicionais (Operador Ternário)**

(condição)?verdade:falso

Ex:

int idade = 18;  
String texto = (idade >= 18) ? "Maior Idade" : "Menor Idade;";  
System.*out*.println("Resultado: " + texto);

**Estruturas condicionais (Switch)**

Switch(valor a testar) {

case valor:

código a ser executado;

break;

}

Caso não seja colocado o break, O código irá executar todos os outros códigos.

**Arrays**



Os Arrays basicamente armazenam vários valores em posições, como se fosse uma lista de outras linguagens. No JAVA é para declarar um array é preciso colocar colchetes no tipo da variável.  
Ex:

String[] nomes = {"Paulo", "Jamilton", "Biro", "Anna"};

int[] numeros = {200, 500, 800, 1000};

**Arrays Multidimensionais**

É basicamente um array dentro do outro. Declaramos eles com dois colchetes ao lado do tipo da variável.

Ex:

String[][] conversas = {  
 {"João", "Olá, tudo bem"}, //0  
 {"Maria", "Tudo certo"}, //1  
 {"Pedro", "Não vi"}, //2  
};

Exemplo simplificado do array anterior

0 1  
0 João Olá, tudo bem  
1 Maria Tudo certo  
2 Pedro Não vi

**Estruturas de repetições (Loops)**

**While:**

While(condição) {

Executa, enquanto a condição for verdadeira.

}

**Forma de incremento**

Ex:

int numero = 10;  
numero = numero + 1;

numero += 1;

numero++;

**FOR**

O for é uma estrutura de repetição bem completa, pois ela já cria a variável inicializadora, condição e o incremento.

Ex:

for(int i = 0; i < 5; i++){

Código para execução

}

**DO WHILE**

O Do While é uma estrutura que faz primeiro uma execução e depois irá fazer o teste logico, sendo útil para programas que irá precisar colocar um ou dois itens fixos e o resto variável.

Ex:

do {

Código para execução

} while(condição);