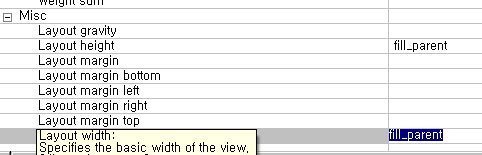
아이디부여

android:id=”@+id/name” 지정하면 R.id.name 상수로 지정됨

findVewById로 호출가능

**layout\_width , layout\_height**



뷰의 폭과높이를 지정

|  |  |
| --- | --- |
| Fill\_parent | 부모의 주어진 크기 다체움 |
| Wrap\_content | 내용물의 크기만큼 채운다 |
| 정수크기 | 지정크기  Px(픽셀) in(인치) mm(밀리미터) pt(포인트)  dp(해상도에 독립적) sp(폰트가변크기) |

**Visibility**

|  |  |
| --- | --- |
| Visible | 보임 |
| Invisible | 숨갸진상태되 자리 차지 |
| Gone | 숨겨지며 자리차지않해 |

**Clickable, longClickable**

마우스클릭을 받을것인지 롱클릭이벤트를 받을것인지 지정

**Focusable**

키보드 포커스를 받을수 있는지 지정.

**다른xml쪽에서 가져오기.**

Ex)String..

@string/id 식..

<TextView

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"wrap\_content"*

android:text=*"@string/hello"*

android:id=*"@+id/mytextview"*/>

</LinearLayout>

**ImageView**

<ImageView android:id=*"@+id/myImg"*

android:src=*"@drawable/icon"*

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"wrap\_content"*

android:tint=*"#4000ff00"/*>

*//살짝 색을입힌다. Src가 레퍼런스니간이 이미지 불러다 쓰는곳은 다 이렇게됨*

버튼 이벤트 리스너 등록

**public** **class** **tt** **extends** **Activity** {

/\*\* Called when the activity is first created. \*/

@Override

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

**setContentView**(**R**.**layout**.*main*);

**Button** btn = (**Button**)**findViewById**(**R**.**id**.*btn*);

btn.**setOnClickListener**(**new** **Button**.**OnClickListener**() {

**public** **void** **onClick**(**View** v) {

**EditText** edit = (**EditText**)**findViewById**(**R**.**id**.*edit*);

**String** str = edit.**getText**().**toString**();

**Toast**.*makeText*(**ButtonEdit**.**this**, str, **Toast**.*LENGTH\_SHORT*).show();

}

});

}

}

**Layout**

<LinearLayout xmlns:android=*"http://schemas.android.com/apk/res/android"*

android:orientation=*"vertical"*

android:gravity=*"bottom"*

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"fill\_parent"*>

</LinearLayout>

android:orientation=*"vertical"* 수직.

android:orientation=*"horizontal" 수평*

레이아웃의 정렬방법을 전체적 정렬방법을 제시한다

android:gravity=*"bottom" 속성*

|  |  |
| --- | --- |
| Center\_horizontal | 수평으로 중앙배치 |
| Left | 컨테이너의 왼쪽배치 크기바뀌지않음 |
| Right | 컨테이너의 오른쪽배치 |
| Fill\_horizontal | 수평방향 가득채움 |
| Center\_vertical | 수직중앙배치 |
| Top | 컨테이너 상단배치 크기바뀌지않음 |
| Bottom | 컨테이너 하단 |
| Fill\_vertical | 수직방향. 가득 |
| Center | 수평으로 수직 중앙배치 |
| Fill | 컨테이너 가득 채우도록 수직수평. |

**베이스 정렬**

Gravity로..

**차일드 영역분할..**

137

**등등…**

**실행도중에 바꾸기**

LinearLayout.setOrientation(). 등등. 으로 정의할수있음

TextView.setGravity()..등등.

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

**setContentView**(**R**.**layout**.layout\_codelayout);

**LinearLayout** MyLinear=(**LinearLayout**)**findViewById**(**R**.**id**.MyLinear);

MyLinear.**setOrientation**(**LinearLayout**.*HORIZONTAL*);

**Button** MyBtn = (**Button**)**findViewById**(**R**.**id**.MyButton);

MyBtn.**setTextSize**(40);

**EditText** MyEdit = (**EditText**)**findViewById**(**R**.**id**.MyEdit);

MyEdit.**setBackgroundColor**(0xff00ff00);

}

**직접적 레이아웃.코딩**

**LinearLayout** linear = **new** **LinearLayout**(**this**);

linear.**setOrientation**(**LinearLayout**.*VERTICAL*);

linear.**setBackgroundColor**(**Color**.*WHITE*);

**TextView** text = **new** **TextView**(**this**);

text.**setText**("TextView");

text.**setGravity**(**Gravity**.*CENTER*);

text.**setTextColor**(**Color**.*RED*);

text.**setTextSize**(20);

linear.**addView**(text);

**setContentView**(linear);

**직접 xml 을 전계하자 (중요)**

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

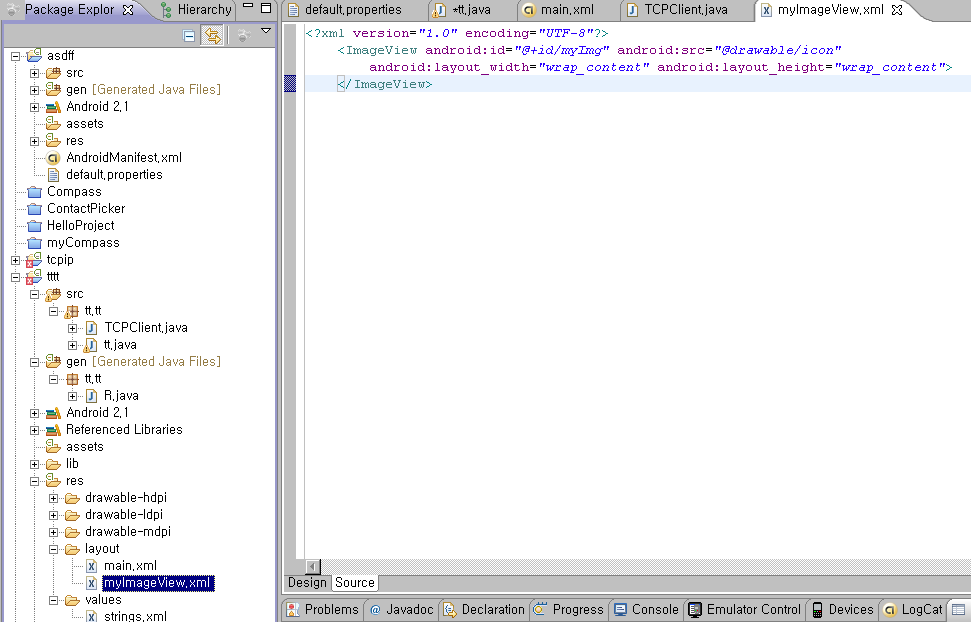
**LayoutInflater** inflater = (**LayoutInflater**)**getSystemService**(**Context**.*LAYOUT\_INFLATER\_SERVICE*);

**LinearLayout** linear = (**LinearLayout**)inflater.**inflate**(**R**.**layout**.*main*, **null**);

**setContentView**(linear);

}

**하나의 컨테이너를 만들어놓고 여기저기서 사용해서 불러쓰자**



저장해놓고 불러다쓰자.

LinearLayout linear = new LinearLayout(this);

linear.setOrientation(LinearLayout.VERTICAL);

linear.setBackgroundColor(Color.WHITE);

**TextView text = (TextView)View.inflate(this, R.layout.myImageView, null);**

linear.addView(text);

setContentView(linear);

**테이블레이아웃 tablelayout**

<TableLayout android:layout\_width=*"fill\_parent"* android:id=*"@+id/tableLayout1"* android:layout\_height=*"wrap\_content"*>

<TableRow android:layout\_height=*"wrap\_content"* android:id=*"@+id/tableRow2"* android:layout\_width=*"wrap\_content"*>

<Button android:layout\_width=*"wrap\_content"* android:id=*"@+id/Button02"* android:layout\_height=*"wrap\_content"* android:text=*"Button"* android:layout\_weight=*"1"*></Button>

<Button android:layout\_width=*"wrap\_content"* android:id=*"@+id/Button01"* android:layout\_height=*"wrap\_content"* android:text=*"Button"* android:layout\_weight=*"1"*></Button>

<Button android:text=*"Button"* android:id=*"@+id/button1"* android:layout\_width=*"wrap\_content"* android:layout\_height=*"wrap\_content"* android:layout\_weight=*"1"*></Button>

</TableRow>

<TableRow android:id=*"@+id/tableRow1"* android:layout\_height=*"wrap\_content"* android:layout\_width=*"wrap\_content"*>

<Button android:layout\_width=*"wrap\_content"* android:id=*"@+id/Button05"* android:layout\_height=*"wrap\_content"* android:layout\_weight=*"1"* android:text=*"j"*></Button>

<Button android:layout\_width=*"wrap\_content"* android:id=*"@+id/Button04"* android:layout\_height=*"wrap\_content"* android:text=*"Button"* android:layout\_weight=*"1"*></Button>

<Button android:layout\_width=*"wrap\_content"* android:id=*"@+id/Button03"* android:layout\_height=*"wrap\_content"* android:text=*"Button"* android:layout\_weight=*"1"*></Button>

</TableRow>

</TableLayout>

Tablerow의 높이는 항상 wrap\_content로..

Tablerow의 가로는 항상 Fill\_parent로..

**Canvas**

190

**입력**

대표적인 콜백 메서드.

|  |  |
| --- | --- |
| onKeyDown | 또한… 키눌렸을때..D패드,키보드, 전화끊기 버튼 통화버튼 뒤로가기버튼 캄메라버튼등… |
| onKeyUp | 사용자가 눌렀던 키를 놓을 때 |
| onTrackballEvent | 트랙볼움직일때 |
| onTouchEvent | 터치스크린눌렀을때 땔 때 이동감지될 때 |

**대표적인 콜백 이벤트 리스너 등록**

**//\* 1.핸들러 메소드 재정의 - 상속을 받아야만 재정의 가능하다.**

**public** **class** **tt** **extends** **Activity** {

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

**View** vw = **new** **MyView**(**this**);

**setContentView**(vw);

}

**protected** **class** **MyView** **extends** **View** {

**public** **MyView**(**Context** context) {

**super**(context);

}

**public** **boolean** **onTouchEvent**(**MotionEvent** event) {

**super**.**onTouchEvent**(event);

**if** (event.**getAction**() == **MotionEvent**.*ACTION\_DOWN*) {

**Toast**.*makeText*(**tt**.**this**,"Touch Event Received",

**Toast**.*LENGTH\_SHORT*).**show**();

**return** **true**;

}

**return** **false**;

}

}

}

**/\* 2.인터페이스 구현 객체 생성 - 별도의 클래스가 필요함**

**public** **class** **tt** **extends** **Activity** {

// 2.리스너 객체 생성

**TouchListenerClass** TouchListener = **new** **TouchListenerClass**();

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

**View** vw = **new** **View**(**this**);

// 3.리스너 등록

vw.**setOnTouchListener**(TouchListener);

**setContentView**(vw);

}

// 1.리스너 구현 클래스 선언

**class** **TouchListenerClass** **implements** **View**.OnTouchListener {

**public** **boolean** **onTouch**(**View** v, **MotionEvent** event) {

**if** (event.**getAction**() == **MotionEvent**.*ACTION\_DOWN*) {

**Toast**.*makeText*(**tt**.**this**,"Touch Event Received",

**Toast**.*LENGTH\_SHORT*).**show**();

**return** **true**;

}

**return** **false**;

}

}

}

1리스너 구현하는 클래스 선언 추상메서드 구현

2 리스너 객체 선언 및 생성

3 준비된 리스너 객체 뷰어 이벤트 연결

**/\* 3.액티비티가 리스너 구현**

**public** **class** **tt** **extends** **Activity** **implements** **View**.OnTouchListener{

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

**View** vw = **new** **View**(**this**);

vw.**setOnTouchListener**(**this**);

**setContentView**(vw);

}

**public** **boolean** **onTouch**(**View** v, **MotionEvent** event) {

**if** (event.**getAction**() == **MotionEvent**.*ACTION\_DOWN*) {

**Toast**.*makeText*(**this**,"Touch Event Received",

**Toast**.*LENGTH\_SHORT*).**show**();

**return** **true**;

}

**return** **false**;

}

}

독립성이 떨어지는 위에소스이다. ..훔 안좋은방법ㅇ이랄까.

/\* 4.뷰에서 리스너 구현

**public** **class** **tt** **extends** **Activity** {

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

**MyView** vw = **new** **MyView**(**this**);

vw.**setOnTouchListener**(vw);

**setContentView**(vw);

}

**protected** **class** **MyView** **extends** **View** **implements** **View**.OnTouchListener {

**public** **MyView**(**Context** context) {

**super**(context);

}

**public** **boolean** **onTouch**(**View** v, **MotionEvent** event) {

**if** (event.**getAction**() == **MotionEvent**.*ACTION\_DOWN*) {

**Toast**.*makeText*(**tt**.**this**,"Touch Event Received",

**Toast**.*LENGTH\_SHORT*).**show**();

**return** **true**;

}

**return** **false**;

}

}

}

**/\* 5.익명 이너 클래스 사용**

**public** **class** **tt** **extends** **Activity** {

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

**View** vw = **new** **View**(**this**);

vw.**setOnTouchListener**(TouchListener);

**setContentView**(vw);

}

**private** **View**.OnTouchListener TouchListener = **new** **View**.**OnTouchListener**() {

**public** **boolean** **onTouch**(**View** v, **MotionEvent** event) {

**if** (event.**getAction**() == **MotionEvent**.*ACTION\_DOWN*) {

**Toast**.*makeText*(**tt**.**this**,"Touch Event Received",

**Toast**.*LENGTH\_SHORT*).**show**();

**return** **true**;

}

**return** **false**;

}

};

}

**/\* 6.익명 이너 클래스의 임시 객체 생성**

**public** **class** **tt** **extends** **Activity** {

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

**View** vw = **new** **View**(**this**);

vw.**setOnTouchListener**( **new** **View**.**OnTouchListener**() {

**public** **boolean** **onTouch**(**View** v, **MotionEvent** event) {

**if** (event.**getAction**() == **MotionEvent**.*ACTION\_DOWN*) {

**Toast**.*makeText*(**tt**.**this**,"Touch Event Received",

**Toast**.*LENGTH\_SHORT*).**show**();

**return** **true**;

}

**return** **false**;

}

});

**setContentView**(vw);

}

}

**외부 레이아웃을 클릭시 안쪽 TextView 내용바꾸기**

**/\* 핸들러내의 지역 변수로 선언하기**

**public** **class** **tt** **extends** **Activity** {

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

**setContentView**(**R**.**layout**.*main*);

**LinearLayout** linear = (**LinearLayout**)**findViewById**(**R**.**id**.*linear*);

linear.**setOnTouchListener**(**new** **View**.**OnTouchListener**() {

**public** **boolean** **onTouch**(**View** v, **MotionEvent** event) {

**if** (event.**getAction**() == **MotionEvent**.*ACTION\_DOWN*) {

**TextView** text = (**TextView**)**findViewById**(**R**.**id**.*text*);

text.**setText**("Touched");

**return** **true**;

}

**return** **false**;

}

});

}

}

**위 같은 방법은 매번 find해야하므로 속도측면에서 아주안좋다.**

**/\* 멤버로 선언해 놓기 매번불러다쓸수있으므로 좋다**

**public** **class** **tt** **extends** **Activity** {

**TextView** mText;

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

**setContentView**(**R**.**layout**.*main*);

mText = (**TextView**)**findViewById**(**R**.**id**.*text*);

**LinearLayout** linear = (**LinearLayout**)**findViewById**(**R**.**id**.*linear*);

linear.**setOnTouchListener**(**new** **View**.**OnTouchListener**() {

**public** **boolean** **onTouch**(**View** v, **MotionEvent** event) {

**if** (event.**getAction**() == **MotionEvent**.*ACTION\_DOWN*) {

mText.**setText**("Touched");

**return** **true**;

}

**return** **false**;

}

});

}

}

위처럼하면 리스너가 많이생기면 생길수록 클래스 지역변수가 생긴다.

**/\* 핸들러를 포함한 메서드의 지역 변수 사용**

**핸들러는 자신을 등록하는 외부 메서드의 지역 변수를 참조해야 하는데 이때 외부 지역변수 final 지정자를 반드시 붙여야 한다.**

**public** **class** **tt** **extends** **Activity** {

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

**setContentView**(**R**.**layout**.*main*);

**final** **TextView** outText = (**TextView**)**findViewById**(**R**.**id**.*text*);

**LinearLayout** linear = (**LinearLayout**)**findViewById**(**R**.**id**.*linear*);

linear.**setOnTouchListener**(**new** **View**.**OnTouchListener**() {

**public** **boolean** **onTouch**(**View** v, **MotionEvent** event) {

**if** (event.**getAction**() == **MotionEvent**.*ACTION\_DOWN*) {

outText.**setText**("Touched");

**return** **true**;

}

**return** **false**;

}

});

}

}

위처럼 하면 메서드 안에 지역변수 이면서도 가져다쓸수있다.

하지만 지역변수기 때문에 onCreate 나가면 사라진다.

**//\* 핸들러를 포함한서브 메서드의 인수 사용**

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

**setContentView**(**R**.**layout**.*main*);

**TextView** outText = (**TextView**)**findViewById**(**R**.**id**.*text*);

**SetTouchHandler**(outText);

}

**void** **SetTouchHandler**(**TextView** tv) {

**final** **TextView** fText = tv;

**LinearLayout** linear = (**LinearLayout**)**findViewById**(**R**.**id**.*linear*);

linear.**setOnTouchListener**(**new** **View**.**OnTouchListener**() {

**public** **boolean** **onTouch**(**View** v, **MotionEvent** event) {

**if** (event.**getAction**() == **MotionEvent**.*ACTION\_DOWN*) {

fText.**setText**("Touched");

**return** **true**;

}

**return** **false**;

}

});

}

**여러가지 이벤트 상황**

onTuchEvent(MotionEvent event);

콜백메서드는 이벤트만 받는다 . 상속하고있기 때문에. 누군지안다 그거지!

onTuch(View v, MotionEvent event);

이해반해 리스너는 View와 event를 받는다 누가 호출했는지 모르니깐 View를 받음

|  |  |
| --- | --- |
| ACTION\_DOWN | 화면을 눌렀다 |
| ACTION\_MOVE | 누른채로 움직였다 |
| ACTION\_UP | 화면에서 산가락을 뗏다 |

**그림판..머리좋은데-\_-훔**

**public** **class** **tt** **extends** **Activity** {

**private** **MyView** vw;

**ArrayList**<**Vertex**> arVertex;

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

vw = **new** **MyView**(**this**);

**setContentView**(vw);

arVertex = **new** **ArrayList**<**Vertex**>();

}

// 정점 하나에 대한 정보를 가지는 클래스

**public** **class** **Vertex** {

**Vertex**(**float** ax, **float** ay, **boolean** ad) {

x = ax;

y = ay;

Draw = ad;

}

**float** x;

**float** y;

**boolean** Draw;

}

**protected** **class** **MyView** **extends** **View** {

**Paint** mPaint;

**public** **MyView**(**Context** context) {

**super**(context);

// Paint 객체 미리 초기화

mPaint = **new** **Paint**();

mPaint.**setColor**(**Color**.*BLACK*);

mPaint.**setStrokeWidth**(3);

mPaint.**setAntiAlias**(**true**);

}

**public** **void** **onDraw**(**Canvas** canvas) {

canvas.**drawColor**(0xffe0e0e0);

// 정점을 순회하면서 선분으로 잇는다.

**for** (**int** i=0;i<arVertex.**size**();i++) {

**if** (arVertex.**get**(i).Draw) {

canvas.**drawLine**(arVertex.**get**(i-1).x, arVertex.**get**(i-1).y,

arVertex.**get**(i).x, arVertex.**get**(i).y, mPaint);

}

}

}

// 터치 이동시마다 정점들을 추가한다.

**public** **boolean** **onTouchEvent**(**MotionEvent** event) {

**if** (event.**getAction**() == **MotionEvent**.*ACTION\_DOWN*) {

arVertex.**add**(**new** **Vertex**(event.**getX**(), event.**getY**(), **false**));

**return** **true**;

}

**if** (event.**getAction**() == **MotionEvent**.*ACTION\_MOVE*) {

arVertex.**add**(**new** **Vertex**(event.**getX**(), event.**getY**(), **true**));

에의해 onDrow가 직간접적으로 호출이 된다

**invalidate**();

**return** **true**;

}

**return** **false**;

}

}

}

**키보드입력**

onKeyDown(int keyCOde, KeyEvent envet) 콜백메서드

View.OnKeyListener 인터페이스 의 Onkey(View v, int keyCode, KeyEvent event)

|  |  |
| --- | --- |
| KEYCODE\_DPAD\_LEFT | 왼쪽이동키 |
| KEYCODE\_DPAD\_RIHGHT | 오른쪽이동키 |
| “ UP | 위쪽 |
| “ DOWN | 아래 |
| “ CENTER | 중앙키 |
| KEYCODE\_A | 알파벳A 알파벳B부터는 KEYCODE\_B식이다. |
| “\_0 | 숫자0.1부터는KEYCODE\_1식이다 |
| “\_CALL | 통화 |
| “\_ENDCALL | 엔드콜 |
| “\_HOME | 홈 |
| “\_BACK | BACK |
| “\_VOLUME\_UP | 볼륨업 |
| “\_VOLUME\_DOWN | 볼륨다운 |

.getAction 메서드는 키보드에 어떤 동작을 했는지 나타내는 세가지값중 하나를 리턴

|  |  |
| --- | --- |
| ACTION\_DOWN | 키눌렀다 |
| ACTION\_UP | 뗏다 |
| ACTION\_MULTIPLE | 같은키 여러 번눌렀다 |

**//onKeyDown(int keyCOde, KeyEvent envet) 콜백메서드**

**public** **class** **tt** **extends** **Activity** {

**private** **MyView** vw;

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

vw = **new** **MyView**(**this**);

vw.**setFocusable**(**true**);

단 키 입력은 포커스를 가진 뷰에게만 전달되므로 생성된 뷰에 포커스는 반드시 주어야 한다

vw.**setFocusableInTouchMode**(**true**);

**setContentView**(vw);

}

**protected** **class** **MyView** **extends** **View** {

**float** mX,mY;

**int** mColor;

**public** **MyView**(**Context** context) {

**super**(context);

mX = 100;

mY = 100;

mColor = **Color**.*BLUE*;

}

**public** **void** **onDraw**(**Canvas** canvas) {

**Paint** Pnt = **new** **Paint**();

canvas.**drawColor**(0xffe0e0e0);

Pnt.**setColor**(mColor);

Pnt.**setAntiAlias**(**true**);

canvas.**drawCircle**(mX,mY,16,Pnt);

}

**public** **boolean** **onKeyDown**(**int** KeyCode, **KeyEvent** event) {

**super**.**onKeyDown**(KeyCode, event);

**if** (event.**getAction**() == **KeyEvent**.*ACTION\_DOWN*) {

**switch** (KeyCode) {

**case** **KeyEvent**.*KEYCODE\_DPAD\_LEFT*:

mX-=5;

**invalidate**();

**return** **true**;

**case** **KeyEvent**.*KEYCODE\_DPAD\_RIGHT*:

mX+=5;

**invalidate**();

**return** **true**;

**case** **KeyEvent**.*KEYCODE\_DPAD\_UP*:

mY-=5;

**invalidate**();

**return** **true**;

**case** **KeyEvent**.*KEYCODE\_DPAD\_DOWN*:

mY+=5;

**invalidate**();

**return** **true**;

**case** **KeyEvent**.*KEYCODE\_DPAD\_CENTER*:

**if** (mColor == **Color**.*BLUE*) {

mColor = **Color**.*RED*;

} **else** {

mColor = **Color**.*BLUE*;

}

**invalidate**();

**return** **true**;

}

}

**return** **false**;

}

}

}

//**View.OnKeyListener 인터페이스 의 Onkey(View v, int keyCode, KeyEvent event) 리스너**

**/\* 임시 객체로 핸들러 만들기**

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

**setContentView**(**R**.**layout**.input\_fruit);

**Button** btnApple=(**Button**)**findViewById**(**R**.**id**.apple);

btnApple.**setOnClickListener**(**new** **Button**.**OnClickListener**() {

**public** **void** **onClick**(**View** v) {

**TextView** textFruit=(**TextView**)**findViewById**(**R**.**id**.fruit);

textFruit.**setText**("Apple");

}

});

**Button** btnOrange=(**Button**)**findViewById**(**R**.**id**.orange);

btnOrange.**setOnClickListener**(**new** **Button**.**OnClickListener**() {

**public** **void** **onClick**(**View** v) {

**TextView** textFruit=(**TextView**)**findViewById**(**R**.**id**.fruit);

textFruit.**setText**("Orange");

}

});

}

**//자기자신에 리스너를 구현**

**/\* 핸들러 통합하기 - 인터페이스 구현**

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

**setContentView**(**R**.**layout**.input\_fruit);

**Button** btnApple=(**Button**)**findViewById**(**R**.**id**.apple);

btnApple.**setOnClickListener**(**this**);

**Button** btnOrange=(**Button**)**findViewById**(**R**.**id**.orange);

btnOrange.**setOnClickListener**(**this**);

}

**public** **void** **onClick**(**View** v) {

**TextView** textFruit=(**TextView**)**findViewById**(**R**.**id**.fruit);

**switch** (v.**getId**()) {

**case** **R**.**id**.apple:

textFruit.**setText**("Apple");

**break**;

**case** **R**.**id**.orange:

textFruit.**setText**("Orange");

**break**;

}

}

**이상적인 방법론. 인터페이스를 오버라이딩한거를 그냥.**

**//\* 핸들러 통합하기 - 인터페이스 구현 멤버 객체**

**public** **class** **Fruit** **extends** **Activity** {

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

**setContentView**(**R**.**layout**.input\_fruit);

**findViewById**(**R**.**id**.apple).setOnClickListener(mClickListener);

**findViewById**(**R**.**id**.orange).setOnClickListener(mClickListener);

}

**Button**.OnClickListener mClickListener = **new** **View**.**OnClickListener**() {

**public** **void** **onClick**(**View** v) {

**TextView** textFruit=(**TextView**)**findViewById**(**R**.**id**.fruit);

**switch** (v.**getId**()) {

**case** **R**.**id**.apple:

textFruit.**setText**("Apple");

**break**;

**case** **R**.**id**.orange:

textFruit.**setText**("Orange");

**break**;

}

}

};

}

**포커스**

터치 모드는 특정뷰나 액티비트가 관리하는게 아니라 시스템 전역적인 상태값이며 사용자의 동작에 따라 모드가 자동으로 바뀐다.

현재 토지모드인가 아닌가는 IsInTouchMode 메서도로 언제든지 조사할수 있다.

|  |  |  |  |
| --- | --- | --- | --- |
| 모드 | XML속성 | 포커수설정 | 포커스 조사 |
| **일반모드** | Focusable | setFocusable | isFocusable |
| **터치모드** | focusableInTouchMode | setFocusableInTouchMode | isFocusableInTouchMode |

키보드를 입력받아야 하는뷰는 반드시 포커스가 가있어야된다

aa.setFocusable(true);

aa.SetFocusableInTouchMode(true)’ 일반 모드와 터치모드에서 포커스를 받을수 있도록 허락하였다

위두개를 설정하지않으면 그쪽에서 포커스를 받지 못한다 그러면 작동이 안되겠죠

264page 봐라. 포커스 이동 건너뛰기등이있다

**타이머**

쓰레드를 쓰는게 원칙이지만 간단한건 타이머로 처리가능하다

안드로이드는 타이머는없지만 대용으로 사용할만한 클래스들이 좀있다.

1초에 한번씩 정수를 증가시켜 텍스트뷰로 출력. 초단위 스톱워치.라고생각해라.

**핸들러로 타이머 효과내기**

**public** **class** **timer** **extends** **Activity** {

/\*\* Called when the activity is first created. \*/

**int** value=0;

**TextView** mText;

@Override

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

**setContentView**(**R**.**layout**.*main*);

mText = (**TextView**)**findViewById**(**R**.**id**.*text*);

mHandler.**sendEmptyMessage**(0);

}

**Handler** mHandler = **new** **Handler**(){

**public** **void** **handleMessage**(android.os.**Message** msg) {

value++;

mText.**setText**("value-"+value);

mHandler.**sendEmptyMessageDelayed**(0, 1000);

};

};

}

* **Handler 클래스는 쓰레드간 메시지를 통신하기 위한 장치로서 sendMessage 메서드나 유사 메서드로 특정 핸들러에게 메시지를 보낼수 있다**
* **handleMessage 로 메시지가 전달된다**
* mHandler.**sendEmptyMessage**(0); 로 빈메시지를 던진다.
* mHandler.**sendEmptyMessageDelayed**(0, 1000); 1초로 후재귀한다

또한 이렇게 내부에 선언하여서 사용할수도있다.

**public** **class** **timer** **extends** **Activity** {

**int** value=0;

**TextView** mText;

**Handler** mHandler;

@Override

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

**setContentView**(**R**.**layout**.*main*);

mText = (**TextView**)**findViewById**(**R**.**id**.*text*);

mHandler = **new** **Handler**(){

**public** **void** **handleMessage**(android.os.**Message** msg) {

value++;

mText.**setText**("value-\*--"+value);

if(value<5)

mHandler.**sendEmptyMessageDelayed**(0, 1000);

};

};

mHandler.**sendEmptyMessage**(0);

}

}

위 클래스는 Handler 클래스 타입이 아니라 handleMessage메서드를 재정의하기위해

Handler로부터 파생되는 익명 클래스 타입이다. 좀더 정확하게 표현하자면.

**mHandler2** handler2 = **new** **mHandler2**();

**class** **mHandler2** **extends** **Handler**{

**public** **void** **handleMessage**(android.os.**Message** msg) {

value++;

mText.**setText**("value---"+value);

handler2.**sendEmptyMessageDelayed**(0, 1000);

};

}

**타이머 클래스**

**public** **class** **timer** **extends** **Activity** {

**int** value = 0;

**TextView** mText;

**Handler** mHandler;

@Override

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

**setContentView**(**R**.**layout**.*main*);

mText = (**TextView**) **findViewById**(**R**.**id**.*text*);

**new** **CountDownTimer**(10 \* 1000, 1000) {

@Override

**public** **void** **onTick**(**long** millisUntilFinished) {

value++;

mText.**setText**("value== " + value);

**if** (value == 5) {

**cancel**();

}

}

@Override

**public** **void** **onFinish**() {

mText.**setText**("value== Finish");

}

}.**start**();

}

}

설명

**new** **CountDownTimer**(10 \* 1000, 1000)

첫번째 인자는 카운트다운할 총시간 두번째 인수는 한번 카운트할 주기이다 1/1000 초이다

**onTick가 호출된다 카운트다운할 총시간이지나면 onFinish가 호출된다**

**메뉴**

272참조

**package** exam.Input;

**import** android.app.\*;

**import** android.os.\*;

**import** android.view.\*;

**import** android.widget.\*;

**import** exam.AndroidExam.\*;

**public** **class** **OptionMenu** **extends** **Activity** {

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

**TextView** MyText = **new** **TextView**(**this**);

MyText.**setText**("메뉴 키를 누르세요.");

**setContentView**(MyText);

}

**public** **boolean** **onCreateOptionsMenu**(Menu menu) {

**super**.**onCreateOptionsMenu**(menu);

MenuItem item=menu.**add**(0,1,0,"짜장");

item.**setIcon**(**R**.**drawable**.*icon*);

item.**setAlphabeticShortcut**('a');

//menu.add(0,1,0,"짜장").setIcon(R.drawable.icon).setAlphabeticShortcut('a');

menu.**add**(0,2,0,"짬뽕").**setIcon**(**R**.**drawable**.*icon*);

SubMenu etc = menu.**addSubMenu**("기타");

etc.**add**(0,3,0,"우동");

etc.**add**(0,4,0,"만두");

**return** **true**;

}

**public** **boolean** **onOptionsItemSelected**(MenuItem item) {

**switch** (item.**getItemId**()) {

**case** 1:

**Toast**.*makeText*(**this**,"짜장은 달콤해",**Toast**.*LENGTH\_SHORT*).**show**();

**return** **true**;

**case** 2:

**Toast**.*makeText*(**this**,"짬뽕은 매워",**Toast**.*LENGTH\_SHORT*).**show**();

**return** **true**;

**case** 3:

**Toast**.*makeText*(**this**,"우동은 시원해",**Toast**.*LENGTH\_SHORT*).**show**();

**return** **true**;

**case** 4:

**Toast**.*makeText*(**this**,"만두는 공짜야",**Toast**.*LENGTH\_SHORT*).**show**();

**return** **true**;

}

**return** **false**;

}

}

**게임**

**public** **class** **timer** **extends** **Activity** {

**GameView** gv;

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

gv = **new** **GameView**(**this**);

**setContentView**(gv);

}

//도형 하나에 대한 정보

**class** **Shape** {

**final** **static** **int** *RECT* = 0;

**final** **static** **int** *CIRCLE* = 1;

**final** **static** **int** *TRIANGLE* = 2;

**int** what;

**int** color;

**Rect** rt;

}

// 게임 뷰 : 실질적인 메인이다.

**class** **GameView** **extends** **View** {

// 그리기 모드. 빈화면 또는 도형 출력

**final** **static** **int** *BLANK* = 0;

**final** **static** **int** *PLAY* = 1;

// 게임 진행 속도

**final** **static** **int** *DELAY* = 1500;

// 현재 그리기 모드

**int** status;

// 생성된 도형의 목록

**ArrayList**<**Shape**> arShape = **new** **ArrayList**<**Shape**>();

**Random** Rnd = **new** **Random**();

**Activity** mParent;

**public** **GameView**(**Context** context) {

**super**(context);

mParent = (**Activity**)context;

// 빈 화면으로 시작하고 잠시 후에 게임 시작

status = *BLANK*;

mHandler.**sendEmptyMessageDelayed**(0,*DELAY*);

}

**public** **void** **onDraw**(**Canvas** canvas) {

// 검정색 배경으로 지운다. 빈 화면이면 지우기만 하고 리턴

canvas.**drawColor**(**Color**.*BLACK*);

**if** (status == *BLANK*) {

**return**;

}

// 도형 목록을 순회하면서 도형 정보대로 출력한다.

**int** idx;

**for** (idx = 0; idx < arShape.**size**(); idx ++) {

**Paint** Pnt = **new** **Paint**();

Pnt.**setColor**(arShape.**get**(idx).color);

**Rect** rt = arShape.**get**(idx).rt;

**switch** (arShape.**get**(idx).what) {

**case** **Shape**.*RECT*:

canvas.**drawRect**(rt, Pnt);

**break**;

**case** **Shape**.*CIRCLE*:

canvas.**drawCircle**(rt.left + rt.**width**()/2, rt.top + rt.**height**()/2,

rt.**width**()/2, Pnt);

**break**;

**case** **Shape**.*TRIANGLE*:

**Path** path = **new** **Path**();

path.**moveTo**(rt.left + rt.**width**()/2, rt.top);

path.**lineTo**(rt.left, rt.bottom);

path.**lineTo**(rt.right, rt.bottom);

canvas.**drawPath**(path, Pnt);

**break**;

}

}

}

**public** **boolean** **onTouchEvent**(**MotionEvent** event) {

**if** (event.**getAction**() == **MotionEvent**.*ACTION\_DOWN*) {

**int** sel;

sel = **FindShapeIdx**((**int**)event.**getX**(), (**int**)event.**getY**());

// 빈 바닥을 찍었으면 무시한다.

**if** (sel == -1) {

**return** **true**;

}

// 마지막 추가된 도형을 제대로 찍었으면 다음 단계로 진행.

// 빈 화면 잠시 보여준 후 새 도형 추가

**if** (sel == arShape.**size**()-1) {

status = *BLANK*;

**invalidate**();

mHandler.**sendEmptyMessageDelayed**(0,*DELAY*);

// 엉뚱한 도형을 찍었으면 질문 후 게임 종료 또는 재시작

} **else** {

**new** **AlertDialog**.**Builder**(**getContext**())

.**setMessage**("재미있지! 또 할래?")

.**setTitle**("게임 끝")

.**setPositiveButton**("함더", **new** DialogInterface.**OnClickListener**() {

**public** **void** **onClick**(DialogInterface dialog, **int** whichButton) {

arShape.**clear**();

status = *BLANK*;

**invalidate**();

mHandler.**sendEmptyMessageDelayed**(0,*DELAY*);

}

})

.**setNegativeButton**("안해", **new** DialogInterface.**OnClickListener**() {

**public** **void** **onClick**(DialogInterface dialog, **int** whichButton) {

mParent.**finish**();

}

})

.**show**();

}

**return** **true**;

}

**return** **false**;

}

// 새로운 도형을 목록에 추가한다.

**void** **AddNewShape**() {

**Shape** shape = **new** **Shape**();

**int** idx;

**boolean** bFindIntersect;

**Rect** rt = **new** **Rect**();

// 기존 도형과 겹치지 않는 새 위치를 찾는다.

**for** (;;) {

// 크기는 32, 48, 64 중 하나 선택

**int** Size = 32 + 16 \* Rnd.**nextInt**(2);

// 위치는 난수로 선택

rt.left = Rnd.**nextInt**(**getWidth**());

rt.top = Rnd.**nextInt**(**getHeight**());

rt.right = rt.left + Size;

rt.bottom = rt.top + Size;

// 화면을 벗어나면 안된다.

**if** (rt.right > **getWidth**() || rt.bottom > **getHeight**()) {

**continue**;

}

// 기존 도형 순회하며 겹치는지 조사한다.

bFindIntersect = **false**;

**for** (idx = 0; idx < arShape.**size**(); idx ++) {

**if** (rt.**intersect**(arShape.**get**(idx).rt) == **true**) {

bFindIntersect = **true**;

}

}

// 겹치지 않을 때 확정한다. 겹치면 계속 새 위치 선정한다.

**if** (bFindIntersect == **false**) {

**break**;

}

}

// 새 도형 정보 작성. 모양, 색상 등을 난수 선택한다.

shape.what = Rnd.**nextInt**(3);

**switch** (Rnd.**nextInt**(5)) {

**case** 0:

shape.color = **Color**.*WHITE*;

**break**;

**case** 1:

shape.color = **Color**.*RED*;

**break**;

**case** 2:

shape.color = **Color**.*GREEN*;

**break**;

**case** 3:

shape.color = **Color**.*BLUE*;

**break**;

**case** 4:

shape.color = **Color**.*YELLOW*;

**break**;

}

shape.rt = rt;

arShape.**add**(shape);

}

// x, y 위치의 도형 번호를 찾는다. 도형이 없는 위치면 -1 리턴

**int** **FindShapeIdx**(**int** x, **int** y) {

**int** idx;

**for** (idx = 0; idx < arShape.**size**(); idx ++) {

**if** (arShape.**get**(idx).rt.**contains**(x, y)) {

**return** idx;

}

}

**return** -1;

}

// 새 도형을 추가하고 화면을 다시 그린다. 시간 딜레이를 주기 위해 핸들러를 사용했다.

**Handler** mHandler = **new** **Handler**() {

**public** **void** **handleMessage**(**Message** msg) {

**AddNewShape**();

status = *PLAY*;

**invalidate**();

**String** title;

title = "MemoryPower - " + arShape.**size**() + " 단계";

mParent.**setTitle**(title);

}

};

}

}

**디버깅**

디버깅을 할려면 먼저 메니페스트에 다음 플래그 지정해야한다

<application android:icon=*"@drawable/icon"* android:label=*"@string/app\_name"* **android:debuggable=*"true"*>**

……

</application>

<uses-sdk android:minSdkVersion=*"7"* />

**DDMS**

디버그 300

**로그**

307

**Adb**

장비상태 관리하는툴

개발장비와 컴퓨터간의 통신 수행 311

**리소스**

코드와 리소스 가 철저하게 분리하되 한발더나아가 리소스와 애셋으로 좀더 세분화 관리한다

* 에셋은(assets) 동영상같은 큰데이터의미
* 리소스는 컴파일되지만 애셋은 원본그대로
* 리소스는 비번사용 되지만 애셋은 상대적 덜빈번

일부 폴더는 만들어져있지만 필요하면 다른것들을 생성해서 사용해야한다

|  |  |  |
| --- | --- | --- |
| 폴더 | 설명 | 읽기메서드 |
| Drawable | 이미지파일도형정의xml파일저장 | getDrawble |
| Layout | 화면의 레이아웃을 저장한다 뷰 그릅과 뷰 파생클래스의 배치 상태가 레이아웃에 기록된다 |  |
| Values | 문자열,색상,배열,크기,단슨드로블,스타일등 여러가지 값들을 정의하는xml파일 | getColor,  getText,  getString,  getDimension |
| Menu | 메뉴구성파일저장한다. |  |
| Xml | 실행중에 읽어서 사용할 임의xml | getXml |
| Raw | 임의의이지낲일저장. | openRawResource |
| Anim | 애니메이션 방식을 정의하는xml파일저장 |  |

|  |  |
| --- | --- |
| 리소스 | 파일명 |
| 문자열 | Strings.xml |
| 색상 | Colors.xml |
| 스타일 | Styles.xml |
| 배열 | Arrays.xml |
| 크기정보 | Dimens.xml |

리소스 컴파일러 appt가 xml 을 소스화시킨다 R.이런걸로.

**리소스정보를 바로 읽어드리자.**

Resource 클래스사용

**public** **class** **timer** **extends** **Activity** {

@Override

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

**setContentView**(**R**.**layout**.widget\_readresource);

**Resources** res = **getResources**();

**TextView** text = (**TextView**)**findViewById**(**R**.**id**.*text*);

**String** str = res.getString(**R**.**string**.textstr);

text.**setText**(str);

//text.setText(R.string.textstr); 위와결과동일하다

**int** textcolor = res.getColor(**R**.color.textcolor);

text.**setTextColor**(textcolor);

//text.setTextColor(R.color.textcolor); 위와결과동일하다

**float** textsize = res.getDimension(**R**.dimen.textsize);

text.**setTextSize**(textsize);

//text.setTextSize(R.dimen.textsize); 위와결과 동일하다

}

}

**Xml 에서 리소스를 참조하자.**

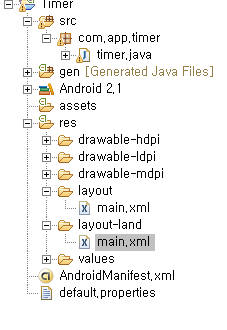
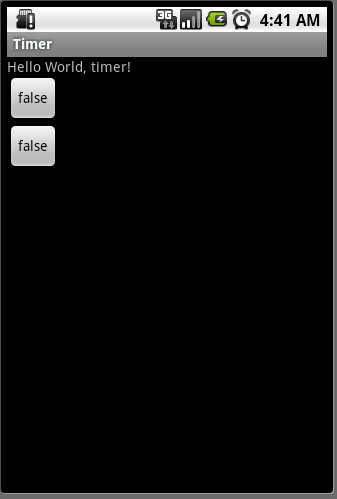
**<TextView android:text=”@string/textstr”/>**

이렇게 “@[패키지:/]타입/id” 사용한다 같은 패키지내에있을땐 패키지명을 생략

|  |  |
| --- | --- |
| 코드에서참조 | R.타입.id |
| 리소스에서 | @타입/id |

**스타일**

324참조



레이아웃을 따로둬서 할수있다.

**텍스트뷰.**

View

EditText

Button

TextView

336참조

* **커서관리 및 선택**
  + 354 텍스트뷰에 포커스왔을 때 커서를 어따가 둘지,. 불록울 잡을지 지정
* **키보드관리**
  + 키보드의 나타나는 종류를 지정할수있다. 358

**소프트키를 보이거나 숨긴다. 예제**

**public** **class** **text** **extends** **Activity** {

**InputMethodManager** mImm;

**EditText** mEdit;

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

**setContentView**(**R**.**layout**.*widget\_showhidekey*);

mImm = (**InputMethodManager**)**getSystemService**(*INPUT\_METHOD\_SERVICE*);

mEdit = (**EditText**)**findViewById**(**R**.**id**.*edit*);

((**Button**)**findViewById**(**R**.**id**.*show*)).**setOnClickListener**(mClickListener);

((**Button**)**findViewById**(**R**.**id**.*hide*)).**setOnClickListener**(mClickListener);

}

**Button**.OnClickListener mClickListener = **new** **Button**.**OnClickListener**() {

**public** **void** **onClick**(**View** v) {

**switch** (v.**getId**()) {

**case** **R**.**id**.*show*:

mImm.**showSoftInput**(mEdit, 0);

**break**;

**case** **R**.**id**.*hide*:

mImm.**hideSoftInputFromWindow**(mEdit.**getWindowToken**(), 0);

**break**;

}

}

};

}

화면에 키보드가 나왔을 때 Activity 화면이 어떻게 보여질지…나타낸다.

360

|  |  |
| --- | --- |
| **WindowSoftInputMode** | **설명** |
| **adjustPan** | **포커스를 받은 뷰가 보이도록 스클홀하여 이동한다 입력받은 뷰는 보이지만 다른뷰들이 가려져 문제가 있다** |
| **adjustResize** | **윈도우 크기를 강제로 조정 뷰의크기가 줄어들기는하지만 다른뷰가 보인다** |
| **adjustUnspecified** | **시스템이 Pan.Resize중하나 자동선택.** |

**나인패치**

P364

**라디오버튼**

**P377**

**이미지버튼.**

**P383**

**어뎁터뷰.**

ListView 는 viewGroup을상속했다.

**public** **class** **text** **extends** **Activity** {

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

**setContentView**(**R**.**layout**.*widget\_listtest*);

//\* 데이터 원본 준비

**ArrayList**<**String**> arGeneral = **new** **ArrayList**<**String**>();

arGeneral.**add**("김유신");

arGeneral.**add**("이순신");

arGeneral.**add**("강감찬");

arGeneral.**add**("을지문덕");

/\* 배열로 준비

String[] arGeneral = {"김유신", "이순신", "강감찬", "을지문덕"};

// 어댑터 준비

**ArrayAdapter**<**String**> Adapter;

Adapter = **new** **ArrayAdapter**<**String**>(**this**, android.**R**.**layout**.*simple\_list\_item\_1*, arGeneral);

// 어댑터 연결

**ListView** list = (**ListView**)**findViewById**(**R**.**id**.*list*);

list.**setAdapter**(Adapter);

}

}

**new** **ArrayAdapter**<**String**>

(**this**, android.**R**.**layout**.*simple\_list\_item\_1*, arGeneral);

현재컨텍스트 액티비트,

레이아웃리소스ID(이미지뷰나 텍스트뷰,체크박스 라이도버스등..)제공하는것도 쓸수있다위처럼

|  |  |
| --- | --- |
| **Simple\_list\_item\_1** | **하나의텍스트뷰로 구성된,** |
| **Simple\_list\_item\_2** | **두개의텍스트뷰,** |
| **Simple\_list\_item.checked** | **오른쪽에 체크표시 나타난다** |
| **Simple\_list\_item\_single\_choice** | **오른쪽에 라디오버튼** |
| **Simple\_list\_item\_muliple\_choice** | **오른쪽에 체크박스.** |

**리소스에 저장해놓고 배열을 불르자!**

**public** **class** **text** **extends** **Activity** {

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

**setContentView**(**R**.**layout**.*widget\_listtest*);

**ArrayAdapter**<CharSequence> Adapter;

Adapter = **ArrayAdapter**.*createFromResource*(**this**, **R**.**array**.*country*,

android.**R**.**layout**.*simple\_list\_item\_1*);

**ListView** list = (**ListView**)**findViewById**(**R**.**id**.*list*);

list.**setAdapter**(Adapter);

}

}

values/arrays.xml

<resources>

<string-array name=*"country"*>

<item>신라</item>

<item>고구려</item>

<item>백제</item>

<item>당나라</item>

<item>발해</item>

<item>옥저</item>

<item>동예</item>

<item>부여</item>

 <item>대한민국</item>

</string-array>

</resources>

**Listview속성.**

list.**setAdapter**(Adapter);

list.**setChoiceMode**(**ListView**.*CHOICE\_MODE\_SINGLE*);

list.**setDivider**(**new** **ColorDrawable**(0xffffff00));

list.**setDividerHeight**(5);

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| choiceMode | 클릭만 가능할뿐 선택할수없지만 이속성은 복수개의항목을 선택할수있다   |  |  |  | | --- | --- | --- | | 속성값 | 인수 | 설명 | | None | CHOICE\_MODE\_NONE | 항목선택할수없음 | | singleChoice | CHOICE\_MODE\_SINGLE | 하나항목선택 | | multipleChoice | CHOICE\_MODE\_MULTIPLE | 복수개 |   복수개나 하나항목은 체크박스나 라디오버튼이있어야한다. |
| Divider | 항목사이에 구분선 지정 |
| dividerHeight | 구분선높이 |
| Entries | 리스트뷰에 표시할 배열지정 |

**항목편집**

**public** **class** **text** **extends** **Activity** {

**ArrayList**<**String**> Items;

**ArrayAdapter**<**String**> Adapter;

**ListView** list;

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

**setContentView**(**R**.**layout**.*widget\_listadddel*);

Items = **new** **ArrayList**<**String**>();

Items.**add**("First");

Items.**add**("Second");

Items.**add**("Third");

Adapter = **new** **ArrayAdapter**<**String**>(**this**, android.**R**.**layout**.

*simple\_list\_item\_single\_choice*, Items);

list = (**ListView**)**findViewById**(**R**.**id**.*list*);

list.**setAdapter**(Adapter);

list.**setChoiceMode**(**ListView**.*CHOICE\_MODE\_SINGLE*);

list.**setOnItemClickListener**(mItemClickListener);

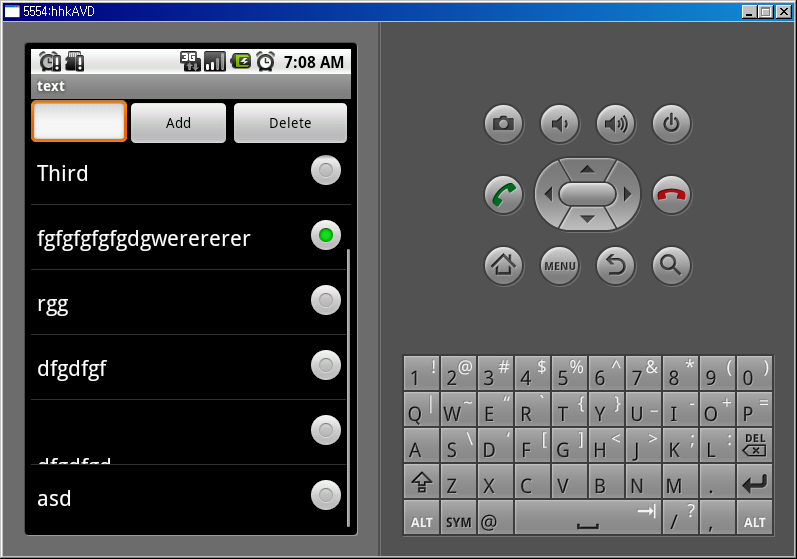
**findViewById**(**R**.**id**.*add*).**setOnClickListener**(mClickListener);

**findViewById**(**R**.**id**.*delete*).**setOnClickListener**(mClickListener);

}

**Button**.OnClickListener mClickListener = **new** **View**.**OnClickListener**() {

**public** **void** **onClick**(**View** v) {

 **EditText** ed = (**EditText**)**findViewById**(**R**.**id**.*newitem*);

**switch** (v.**getId**()) {

**case** **R**.**id**.*add*:

**String** text = ed.**getText**().**toString**();

**if** (text.**length**() != 0) {

Items.**add**(text);

ed.**setText**("");

Adapter.**notifyDataSetChanged**();

}

**break**;

**case** **R**.**id**.*delete*:

**int** id;

id=list.**getCheckedItemPosition**();

**if** (id != **ListView**.*INVALID\_POSITION*) {

Items.**remove**(id);

list.**clearChoices**();

Adapter.**notifyDataSetChanged**();

}

**break**;

}

}

};

**AdapterView**.OnItemClickListener mItemClickListener = **new** **AdapterView**.**OnItemClickListener**() {

@SuppressWarnings("unchecked")

**public** **void** **onItemClick**(**AdapterView** parent, **View** view, **int** position, **long** id) {

**String** mes;

mes = "Select Item = " + Items.**get**(position);

**Toast**.*makeText*(**text**.**this**,mes,**Toast**.*LENGTH\_SHORT*).**show**();

} };}

**커스텀 리스트뷰**

**401**

**여러종류의항목뷰**

**407**

**ListActivity**

**413**

**스피너(여러가지 선택중 하나만선택 받을 때)**

**417**

**그리드 (바둑반처럼…)**

**421**

**이미지뷰(갤러리)**

**426**

**프로그래스 바 (로딩..)**

**requestWindowFeature(Window.FEATURE\_INDETERMINATE\_PROGRESS);**

**setProgressBarIndeterminateVisibility(true);**

**427**

**시크바 (볼륨이나. 동영상 위치등.. 스크롤같은거)**

**래이딩 바 (별포나..)**

**위젯**

**442**

**자료관리**

**486**

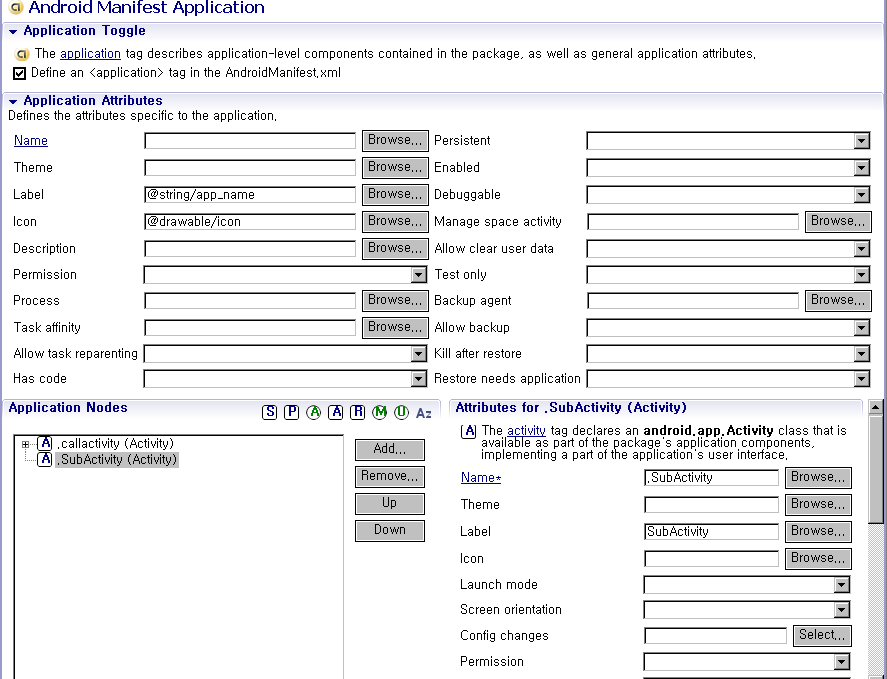
* CP(ContentProvider) 유일하게 응용프로그램끼리 공유..가능한거.526

**URI**

**526**

**액티비티**

* 모든 프로그램에 포함된 모든 액티비티는 반드시 매니페스트에 등록되어야한다
* 등록안되면 없는걸로 간주한다



* **1 새로만들 애티비티의 레이아웃을 xml정의**
* **2 액티브티 코드를java작성**
* **3 tofh 추가한 애티비티 매니페스트 등록**
* **4 startActivity 메서드로 애티비트 호출**

**public** **class** **callactivity** **extends** **Activity** {

/\*\* Called when the activity is first created. \*/

@Override

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

**setContentView**(**R**.**layout**.*main*);

**Button** btncall=(**Button**)**findViewById**(**R**.**id**.*call*);

btncall.**setOnClickListener**(**new** **Button**.**OnClickListener**() {

@Override

**public** **void** **onClick**(**View** v) {

**Intent** intent = **new** **Intent**(**callactivity**.**this**,**SubActivity**.**class**);

**startActivity**(intent);

}

});

}

}

**///////////////////////////////////////////**

**public** **class** **SubActivity** **extends** **Activity** {

@Override

**protected** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

**setContentView**(**R**.**layout**.*subactivity*);

**Button** btnClose = (**Button**)**findViewById**(**R**.**id**.*close*);

btnClose.**setOnClickListener**(**new** **Button**.**OnClickListener**(){

@Override

**public** **void** **onClick**(**View** v) {

**finish**();

}

});

}

}

**startActivity() //애티브티 호출**

**Finish() //close**

**인덴트 549**

- 뭔가 작업을 시키기위해 호출하는 이유다

**- ★애티브티끼리 서로통신 하기위해서 인덴트를 사용한다**

- 인덴트는 액티비티뿐만 아니라 서비스,cp,br 등의컴포넌트들이 수행해야할 작업에 대한

정보를 가지며 작업 결과를 돌려주기 위해서 사용된다

ex) 그림을 보여주는 애티브티를 호출한다면 출력해야할 그림이 무엇인지 알려줘야하며 입력을 받은 액티비트는 사용자가 입력한 정보가 무엇인지 리턴해야한다

액티비트를 호출하는 메서드

* **seartActivity(Intent intent)**

인텐트 하나만을 인수로 취하는데 인텐트 안에는 누구를 호출하여 무슨 작업을시킬것인가 대한 정보가 있다. 그러니 호출전 인텐트를 정의해야한다

Intent()

**Intent(Context, Class<?>)**

Intent(Intent)

Intent(String)

Intent(String, Uri)

Intent(String, Uri, Context, Class<?>)

생성자를 이렇가 가지고있다. 보통 2번째 많이써서 서브 액티브티를 부른다

**Intent intent = new Intent(callactivity.this,SubActivity.class);**

호출자(액티브티), 호출할 애티브티

//권한만있다면 외부 다른 애티브티도 불를수있다

**암시적인 인덴트**

주로 다른 응용프로그램의 액티브티를 호출할 때 사용한다.546

**Action**

실행하고자 하는 동작아며 인텐트를 통해 어떤 작업을 수행할지를 지정한다 BR컴포넌트의 경우 어떤 사건이 발생했는지를 알리기도한다

|  |  |  |
| --- | --- | --- |
| 액션 | 대상 | 설명 |
| ACTION\_CALL | 액티비티 | 통화를 시작한다 |
| ACTION\_EDIT | 액티비티 | 데이터를 표시하고 편집함 |
| ACTION\_MAIN | 액티비티 | 메인애티비트를 실행한다 입출력되는 데이터가없다 |
| ACTION\_VIEW | 액티비티 | 뭔가를 보여준다 |
| ACTION\_DIAL | 액티비티 | 전화를건다 |
| ACTION\_BATTERY\_LOW | BR | 배터리가 부족하다 |
| ACTION\_HEADSET\_PLUG | BR | 헤드셋장비에 접속되거나 분리 |
| ACTION\_SCREEN\_ON | BR | 화면이 켜졌다 |
| ACTION\_ | BR | 타임존이 변경됐다 |

동작의 종류가 워낙다양하다 임의의 동작을 정의할수도 있다.

액션을 조사하거나 변경할때는 getAction, setAction 메서드를 사용한다

**DATA**

동작에 필요한 데이터 제공   
**예)**를들어 ACTION\_EDIT doruts에 대해서는 어떤 파일을 편집할 것인가 지정 해야하며

ACTIOn\_CALL 액션에 대해서는 누구에게 전화를 걸건가를 지정한다.

목적이되는 대상이 광범위하다 그래서 URI로 지정되어있다

데이터를 액세스할 때 **getData,setData메서드를 사용한다**

액션을 처리할 컴포넌트가 애매한경우 아래 정보들로 가급적 상세히 지정해야한다

|  |  |
| --- | --- |
| Type | 데이터의타입은 대게 경우 자동판별 가능  http:// 는 웹, tel: 은 전화번호  확장자가 jpg지만 실제로는 png 포맷일수있으면 아에없거나.  getType, setType 메서드로 데이터 MIME 타입 직접 지정해야한다 |
| Category | 실행할 액션에 대한 좀더 추가적인 상세 정보를 제공한다 추가 정보이므로 여러 개를 동시에 지정가능 addCategory 메서드는 카테고리를 추가하고 removeCategory는 카테고리를 제거한다 |
| Component | 인텐트를 처리할 컴퓨넌트를 명시적으로 지정한다 이속성이 지정되면 명시적 인텐트가 되므로 다른정보들은 더 이상 참조하지않는다 |
| Extras | 그외 컴포넌트로 전달되어야 할 추가적인 정보들이다 키와 값의 쌍으로 저장되어 컴포넌트에 게 전달되며 리턴 용도로도 사용된다 저장되는 타입에 따라putExtra메서드가 여러벌 오버로딩되어 있으며 읽기 메서더는 읽는 타입에 따라 putExtra 메서드가 여러벌 오버로딩되어 있으며 읽기 메서드는 읽는 타입에 따라 getInExtra, getStringExtra등으로 제공된다 이름만 중복되지 않으면 얼마든지 많은 정보를 전달할수 있다 |
| Flags | 액티비티를 띄울 방법이나 액티비티를 관리하는 방법등에 대한 옵션 정보들이 저장된다 setflag 메서드로 플래그 전체를 대입하거나 addflags aptjemfh 특정 플래그만 추가로 더 지정할수 있다  Action,Data 까지는 직관적으로 그 의미를 쉽게 이해할수 있겟지만 Category나 Flag들은 지금당장 의미를 이해하기 힘들다 또한 플래그별로 정확한 효과들까지 파악하는데는 상당한 시간과 경험을 피룡로 한다 본질적으로 양이 많고 본잡하므로 지금은 이해가 되는데까지만 접수해두고 시간이 날때마다 하나씩 각개 격파하자 |

**암시적 인텐트**

앞에선 본건 명시적 인텐트였다

암시적 인텐트를 사용하면 다른 패키지에 속한 임의의 액티비티나 서비스도 호출할수 있다 권한만있다면 특정 패키지의특정 액티비티를 명시적으로 호출하는것도 가능하다

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

**setContentView**(**R**.**layout**.*activity\_callother*);

**findViewById**(**R**.**id**.*web*).**setOnClickListener**(mClickListener);

**findViewById**(**R**.**id**.*dial*).**setOnClickListener**(mClickListener);

**findViewById**(**R**.**id**.*picture*).**setOnClickListener**(mClickListener);

**findViewById**(**R**.**id**.*other*).**setOnClickListener**(mClickListener);

}

**Button**.OnClickListener mClickListener = **new** **View**.**OnClickListener**() {

**public** **void** **onClick**(**View** v) {

**Intent** intent;

**switch** (v.**getId**()) {

**case** **R**.**id**.*web*:

intent = **new** **Intent**(**Intent**.*ACTION\_VIEW*,

**Uri**.*parse*("http://www.google.com"));

**startActivity**(intent);

**break**;

**case** **R**.**id**.*dial*:

intent = **new** **Intent**(**Intent**.*ACTION\_DIAL*, **Uri**.*parse*("tel:015-123-4567"));

**startActivity**(intent);

**break**;

**case** **R**.**id**.*picture*:

intent = **new** **Intent**(**Intent**.*ACTION\_VIEW*);

**Uri** uri = **Uri**.*fromFile*(**new** **File**("/sdcard/test.jpg"));

intent.**setDataAndType**(uri, "image/jpeg");

**startActivity**(intent);

**break**;

**case** **R**.**id**.*other*:

intent = **new** **Intent**(**Intent**.*ACTION\_MAIN*);

intent.**setComponent**(**new** **ComponentName**("exam.Input", "exam.Input.Input"));

//intent.setClassName("exam.Input", "exam.Input.Input");

**startActivity**(intent);

**break**;

}

}

};

552

* 첫번째 뭐를 보여줘라 구굴URI전송
* 두번째 다이얼을 하는데 번호로한다
* 해당jpg를 보여줘라
* 해당 액티브티를 뛰어라. 없으면 오류난다

**액티브티간의 통신 553**

인텐트는 액티비티간의 인수와 리턴값을 전달하는 도구로 사용 된다

이때 주로 Bundle 타입의 Extras를 활용하는데 이름고 ㅏ값의 쌍으로 된 임의 타입의 정보를 원하는 개수 만큼 전달할수 있다

* 넣기

Intent putExtra(String name,int value);

Intent putExtra(String name,String value);

Intent putExtra(String name,boolean value);

* 꺼내기

getIntExtra(String name,int defaultValue)

String getStringExtra(String name)

Boolean getBooleanExtra(String name,Boolean defaultValue);



**public** **class** **callactivity** **extends** **Activity** {

/\*\* Called when the activity is first created. \*/

**TextView** mText;

**final** **static** **int** *ACT\_EDIT* = 0;

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

**setContentView**(**R**.**layout**.*activity\_commactivity*);

mText = (**TextView**)**findViewById**(**R**.**id**.*text*);

**Button** btnEdit=(**Button**)**findViewById**(**R**.**id**.*edit*);

btnEdit.**setOnClickListener**(**new** **Button**.**OnClickListener**() {

**public** **void** **onClick**(**View** v) {

**Intent** intent = **new** **Intent**(**callactivity**.**this**, **ActEdit**.**class**);

intent.**putExtra**("TextIn", mText.**getText**().**toString**());

**startActivityForResult**(intent,*ACT\_EDIT*);

}

});

}

**protected** **void** **onActivityResult** (**int** requestCode, **int** resultCode, **Intent** data) {

**switch** (requestCode) {

**case** *ACT\_EDIT*:

**if** (resultCode == *RESULT\_OK*) {

mText.**setText**(data.**getStringExtra**("TextOut"));

}

**break**;

}

}

}

////////////////////////

**public** **class** **ActEdit** **extends** **Activity** {

**EditText** mEdit;

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

**setContentView**(**R**.**layout**.*activity\_actedit*);

mEdit = (**EditText**)**findViewById**(**R**.**id**.*stredit*);

**Intent** intent = **getIntent**();

mEdit.**setText**(intent.**getStringExtra**("TextIn"));

**Button** btnOK=(**Button**)**findViewById**(**R**.**id**.*ok*);

btnOK.**setOnClickListener**(**new** **Button**.**OnClickListener**() {

**public** **void** **onClick**(**View** v) {

**Intent** intent = **new** **Intent**();

intent.**putExtra**("TextOut", mEdit.**getText**().**toString**());

**setResult**(*RESULT\_OK*,intent);

**finish**();

}

});

**Button** btnCancel=(**Button**)**findViewById**(**R**.**id**.*cancel*);

btnCancel.**setOnClickListener**(**new** **Button**.**OnClickListener**() {

**public** **void** **onClick**(**View** v) {

**setResult**(*RESULT\_CANCELED*);

**finish**();

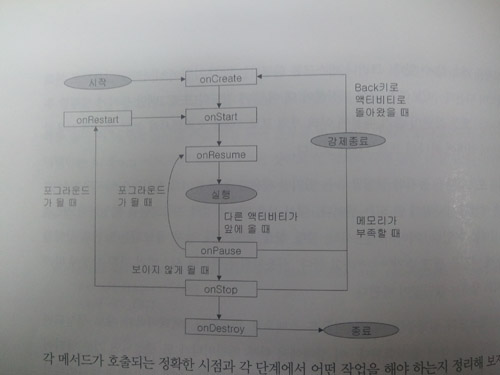
}

});

}

}

**P558참고. 다른 이벤트도있을경우 기능의 불르는기능**

**액티브티생명주기**

560

|  |  |
| --- | --- |
| Oncreate | 액티비티를 촉화한다 중지했다가 재시작하는 경우라면 액티비티의 이전 상태 정보인 번틀이 전달된다 이정보대로 재초기화된다   * 엑티비티가 처음 시작될때 인자로null 값을 넘겨받으면서 호출된다 (시스템재시작경우) * 액티비티가 실행된 적이 있는데 어떤 이유건 종료됐다가 다시실행되는경우 onSaveInstanceState() 메서드로 보관해둔 Bundle 인스턴스를 넘겨받는다, 저장할때는  onRestoreInstanceState(Bundle) * 액티비트가 실행중인데 하드웨어 상태(예를들어 가로체그 세로보기 모드변경)에 따라 다른 시소스를 지정해 개발한 경우 하드웨어 상태가 변경되면 액티비티가 새로 생성돼 실행되면서 onCreate() 메서드 호출된다. |
| onRestart | 액티비트가 재시작될 때 호출된다 특별히 할일은없다 |
| onStart | 액티비트가 사용자에게 보이기 직전에 호출된다 |
| onResume | 사용자와 상호작용을 하기 직전에 호출된다 이단계에서 스택의 제일 위로올라온다  정지상태에서->다시 시작될때 ,  액티비트가 처음 실행될때. |
| onPause | 다른 액티비트가 실행될 때 호출된다 이 단계에서 미저장한 데이터가 있으면 저장하고 애니메이션은 중지해야한다 이메스다가 리턴되어야 새애기비트가 활성화되므로 시간을 너무 많이 끌어선 안된다  전면에서 뒤로 물러설때. |
| onStop | 액티비트가 사용자에게 보이지 않게 될 때 호출한다  액티비트가 정지될때 |
| onDestroy | 액티비티가 파괴될 때 호출된다 시스템에 의해 강제 종료되는것인지 아니면 finish메서드 호출에 의해 스스로 종료되는것인지는 isFinishing 메서드로 조사 가능하다.  Finish() 호출해서 죽는거는 메모리가 부족해서 죽는거다. |
| onSaveInstanceState | 회전돼 실행중인 액티비트가 종료되면 이함수를 통해 번들을 저장한다 |
| onRestoreInstanceState | 메서드는 다시 여기서 번들을 꺼내서 사용한다. |

**탭**

**576**

**대화상자590**

일반 viw뛰우기 팝업으로

**PopupWindow** popup;…

**private** **void** **popupSetting**() {

**View** popupview = **View**.*inflate*(**centerMap**.**this**, **R**.**layout**.*custom\_list*, **null**);

popup = **new** **PopupWindow**(popupview,100,100,**true**);

}…

popup.**showAtLocation**(map, **Gravity**.*CENTER*, 0, 0); //가운데 놓기

**쓰레드630**

**//\* Thread 객체 사용**

**int** mMainValue = 0;

**int** mBackValue = 0;

**TextView** mMainText;

**TextView** mBackText;

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

**setContentView**(**R**.**layout**.*thread\_threadtest*);

mMainText = (**TextView**)**findViewById**(**R**.**id**.*mainvalue*);

mBackText = (**TextView**)**findViewById**(**R**.**id**.*backvalue*);

**Button** btn = (**Button**)**findViewById**(**R**.**id**.*increase*);

btn.**setOnClickListener**(**new** **Button**.**OnClickListener**() {

**public** **void** **onClick**(**View** v) {

mMainValue++;

mMainText.**setText**("MainValue : " + mMainValue);

mBackText.**setText**("BackValue : " + mBackValue);

}

});

**BackThread** thread = **new** **BackThread**();

thread.**setDaemon**(**true**);

thread.**start**();

}

**class** **BackThread** **extends** **Thread** {

**public** **void** **run**() {

**while**(**true**) {

mBackValue++;

//mBackText.setText("BackValue : " + mBackValue);

**try** { **Thread**.*sleep*(1000); } **catch** (**InterruptedException** e) {;}

}

}

}

**/\* Runnable 객체 사용하기**

**int** mMainValue = 0;

**int** mBackValue = 0;

**TextView** mMainText;

**TextView** mBackText;

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

**setContentView**(**R**.**layout**.*thread\_threadtest*);

mMainText = (**TextView**)**findViewById**(**R**.**id**.*mainvalue*);

mBackText = (**TextView**)**findViewById**(**R**.**id**.*backvalue*);

**Button** btn = (**Button**)**findViewById**(**R**.**id**.*increase*);

btn.**setOnClickListener**(**new** **Button**.**OnClickListener**() {

**public** **void** **onClick**(**View** v) {

mMainValue++;

mMainText.**setText**("MainValue : " + mMainValue);

mBackText.**setText**("BackValue : " + mBackValue);

}

});

**BackRunnable** runnable = **new** **BackRunnable**();

**Thread** thread = **new** **Thread**(runnable);

thread.**setDaemon**(**true**);

thread.**start**();

}

**class** **BackRunnable** **implements** Runnable {

**public** **void** **run**() {

**while** (**true**) {

mBackValue++;

**try** { **Thread**.*sleep*(1000); } **catch** (**InterruptedException** e) {;}

}

}

}

thread.**setDaemon**(**true**);\

무한루푸이기 때문에 주쓰레드가 죽으면 나의쓰레드도 같이 죽으라는 명령이다.

**위코드는 버튼을 눌러야지만 나타난다. 이건 부적절한 코딩일수있다**

변경을하자 ///// **하지만 이코드는 에러가난다!!!!!!!**

**class** **BackRunnable** **implements** Runnable {

**public** **void** **run**() {

**while** (**true**) {

mBackValue++;

mBackText.**setText**("BackValue : " + mBackValue);

**try** { **Thread**.*sleep*(1000); } **catch** (**InterruptedException** e) {;} }}}

**백그라운드 스레드는 내부적인 연산만 해야 하며 다른 스레드 소속의UI를 건들수없다**

**왜냐면 한 개ㅑㄱ체를 두쓰레드가 동시에 변경 허용 안한다! 동기화문제 때문에**

**쓰레드간 통신을 하자 Handler**

Handler 클래스로 표현되는 핸들러는 쓰레드간 **메시지나, 러너블** 객체를 통해 메시지를 주고받는 장치이다. **//메시지는 큐로 저장된다**

Public void handleMessage(Message msg)

인수로 메시지(Message)객체 하나를 전달받는데 메시지는 스레드간 통신 내용을 저장하는 객체이다 단순한 신호나 명령뿐만 아니라 복잡한 추가 정보도 전달받아야 하므로 여러 개의 필드들을 가진다 각필드의 구체적인 값은 메시지마다 다르다

|  |  |
| --- | --- |
| 필드 | 설명 |
| Int what | 메시지의 의미를 설명한다 의미가 정해져 있지 않으며 sgosemffjquffh 지역적이므로 다른 핸들러와 충돌 위험없다 |
| Int arg1 | 메시지의 추가정보이다 |
| Int arg2 | 메시지의 추가정보이다 |
| Object obj | 정수만으로 메시지를 기술할수 없을 때 임의의 객체를 보낸다 |
| Messenger replyTo | 메시지에 대한 응답을 받을 객체를 지정한다 |

메시지(Message)를 보낼 때 다음 메서드를 사용한다

|  |  |
| --- | --- |
| boolean Handler.sendEmptyMessage(int what) | What 값만 전달할 때 간단하게 사용 |
| Boolean Handler.sendMessage(Message msg) | 위보다 복잡한내용보낼 때. 어떤 사실이나 지시만 통보할때는 추가 정보가 없는 빈메시지를 쓰는 것이 편리 |
| Boolean sendMessageAtFrontQfQueue(Message msg) | 큐의 제일 앞쪽에 놓아 우선적으로 처리할수도있다. |

**int** mMainValue = 0;

**int** mBackValue = 0;

**TextView** mMainText;

**TextView** mBackText;

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

**setContentView**(**R**.**layout**.*thread\_threadtest*);

mMainText = (**TextView**)**findViewById**(**R**.**id**.*mainvalue*);

mBackText = (**TextView**)**findViewById**(**R**.**id**.*backvalue*);

**Button** btn = (**Button**)**findViewById**(**R**.**id**.*increase*);

btn.**setOnClickListener**(**new** **Button**.**OnClickListener**() {

**public** **void** **onClick**(**View** v) {

mMainValue++;

mMainText.**setText**("MainValue : " + mMainValue);

}

});

**BackThread** thread = **new** **BackThread**();

thread.**setDaemon**(**true**);

thread.**start**();

}

**class** **BackThread** **extends** **Thread** {

**public** **void** **run**() {

**while** (**true**) {

mBackValue++;

mHandler.**sendEmptyMessage**(0);

**try** { **Thread**.*sleep*(1000); } **catch** (**InterruptedException** e) {;}

}

}

}

**Handler** mHandler = **new** **Handler**() {

**public** **void** **handleMessage**(**Message** msg) {

**if** (msg.what == 0) {

mBackText.**setText**("BackValue : " + mBackValue);

}

}

};

타이머와 비슷하게 구현이된다

메인스레드는 mHandler를 멤버로 정의하고 0번 메시지를 받았을 때 mBackText에 mBackValue값을 출력한다.

mHandler.**sendEmptyMessage**(0); 로 호출을하고. 핸들러에서

분기처리를 위해 ( 다른 쓰레드에서 사용하는걸 구분에 따라서 다른 작업을위해)

**Handler** mHandler = **new** **Handler**() {

**public** **void** **handleMessage**(**Message** msg) {

**if** (msg.what == 0) {

mBackText.**setText**("BackValue : " + mBackValue);

}

}

};

를한다.

**boolean post (Runable r) 쓰레드 안에서 핸들사용할수있다. 변수공유 쉬워짐 소스 감편해짐**

메시지는 어떤 작업을 하라는 지시사항을 보내는 것인데 비해 러너블 작업을 직접 수행하는 코드라는점이 다르다 간단한 작업의 경우는 그냥 코드를 보내는게 좋다

런너블 (**Runnable**)객체를 던진다.

**/\* Runnable 객체 보내기**

**public** **class** **callactivity** **extends** **Activity** {

**int** mMainValue = 0;

**int** mBackValue = 0;

**TextView** mMainText;

**TextView** mBackText;

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

**setContentView**(**R**.**layout**.*thread\_threadtest*);

mMainText = (**TextView**)**findViewById**(**R**.**id**.*mainvalue*);

mBackText = (**TextView**)**findViewById**(**R**.**id**.*backvalue*);

**Button** btn = (**Button**)**findViewById**(**R**.**id**.*increase*);

btn.**setOnClickListener**(**new** **Button**.**OnClickListener**() {

**public** **void** **onClick**(**View** v) {

mMainValue++;

mMainText.**setText**("MainValue : " + mMainValue);

}

});

**BackThread** thread = **new** **BackThread**();

thread.**setDaemon**(**true**);

thread.**start**();

}

**class** **BackThread** **extends** **Thread** {

**public** **void** **run**() {

**while** (**true**) {

mBackValue++;

mHandler.**post**(**new** **Runnable**() {

**public** **void** **run**() {

mBackText.**setText**("BackValue : " + mBackValue);

}

});

**try** { **Thread**.*sleep*(1000); } **catch** (**InterruptedException** e) {;}

}

}

}

**Handler** mHandler = **new** **Handler**();

}

핸들러는 객체만있으면된다

**외부 쓰레드를 사용하자**

**전형적인 방법** **/\* 외부 스레드 리턴값을받자!!**

**public** **class** **callactivity** **extends** **Activity** {

**int** mMainValue = 0;

**TextView** mMainText;

**TextView** mBackText;

**BackThread** mThread;

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

**setContentView**(**R**.**layout**.*thread\_threadtest*);

mMainText = (**TextView**)**findViewById**(**R**.**id**.*mainvalue*);

mBackText = (**TextView**)**findViewById**(**R**.**id**.*backvalue*);

**Button** btn = (**Button**)**findViewById**(**R**.**id**.*increase*);

btn.**setOnClickListener**(**new** **Button**.**OnClickListener**() {

**public** **void** **onClick**(**View** v) {

mMainValue++;

mMainText.**setText**("MainValue : " + mMainValue);

}

});

mThread = **new** **BackThread**(mHandler);

mThread.**setDaemon**(**true**);

mThread.**start**();

}

**Handler** mHandler = **new** **Handler**() {

**public** **void** **handleMessage**(**Message** msg) {

**if** (msg.what == 0) {

mBackText.**setText**("BackValue : " + msg.arg1);

}

}

};

}

독립성가진 쓰레드기 때문에

다른애티브티에서 사용가능하다

특정 처리라면 이전예제처럼해라

**class** **BackThread** **extends** **Thread** {

**int** mBackValue = 0;

**Handler** mHandler;

**BackThread**(**Handler** handler) {

mHandler = handler;

}

**public** **void** **run**() {

**while** (**true**) {

mBackValue++;

**Message** msg = **new** **Message**();

msg.what = 0;

msg.arg1 = mBackValue;

mHandler.**sendMessage**(msg);

**try** { **Thread**.*sleep*(1000); } **catch** (**InterruptedException** e) {;} } }}

**정보를 주고받을 때 Message객체가 필요하다**

**1 별도의 쓰레드 객체이다,2 핸들러를 넘겨야한다 ,3 쓰레드에서 핸들러에 값을넣고 리턴을한다**

**Arg1,art2 로 2가지정보를보낼수있지만 더많은정보보낼려면 obj멤버를 사용해라**

**매번Message객체를 생성해서 데이터를 주고받으면 (new)속도가 느리다 따라서**

**안드로이에서는 풀poll을 제공한다 메시지풀을 메시지 객체를 꺼낼 때 다음메서드 사용**

**static Message obtain(..)**

**static Message obtain (Handler h, int arg1,int arg2,object obj);**

**void recycle() 사용한메시지를 풀에 다시 집어넣는다 그건 시스템이 관리**

* **obtain은 메서드는 메시지 풀에서 비슷한 메시지를 꺼내 재사용한다**

**호출하면 객체하나를 꺼내준다Message**

**//\* 외부 스레드 - obtain으로 메시지 얻기**

**public** **class** **callactivity** **extends** **Activity** {

**int** mMainValue = 0;

**TextView** mMainText;

**TextView** mBackText;

**BackThread** mThread;

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

**setContentView**(**R**.**layout**.*thread\_threadtest*);

mMainText = (**TextView**)**findViewById**(**R**.**id**.*mainvalue*);

mBackText = (**TextView**)**findViewById**(**R**.**id**.*backvalue*);

**Button** btn = (**Button**)**findViewById**(**R**.**id**.*increase*);

btn.**setOnClickListener**(**new** **Button**.**OnClickListener**() {

**public** **void** **onClick**(**View** v) {

mMainValue++;

mMainText.**setText**("MainValue : " + mMainValue);

}

});

mThread = **new** **BackThread**(mHandler);

mThread.**setDaemon**(**true**);

mThread.**start**();

}

**Handler** mHandler = **new** **Handler**() {

**public** **void** **handleMessage**(**Message** msg) {

**if** (msg.what == 0) {

mBackText.**setText**("BackValue : " + msg.arg1);

}

}

};

}

**class** **BackThread** **extends** **Thread** {

**int** mBackValue = 0;

**Handler** mHandler;

**BackThread**(**Handler** handler) {

mHandler = handler;

}

**public** **void** **run**() {

**while** (**true**) {

mBackValue++;

**Message** msg = **Message**.*obtain*(mHandler, 0, mBackValue, 0);

mHandler.**sendMessage**(msg);

**try** { **Thread**.*sleep*(1000); } **catch** (**InterruptedException** e) {;}

}

}

}

**루퍼**

메시지는 즉시호출되는 메서드와달리 보내는 신호이므로 바로 전송이 안될수있다.

어디엔가 쌓아놓다가 출격한다.(메시지큐)

**-메시지나 러너블 객체는 일단 큐에 저장되고 들어온 순서대로 순차적으로 처리된다**

* **메시지큐에서 핸들러로 전달하는것인 바로 루퍼가 한다.**
* **메인쓰레드는 기본적으로 루퍼를 가진다**
* **다른 필요로의해 만들어진 쓰레드는 기본적으로 루퍼를 가지지않는다**
* **핸들러를 만들어놓고 전달되는 메시지를 처리하기만 하면된다**
* **루퍼 직접프로그래밍할때는 작업쓰레드가 메시지를 받아야 할때다.**

**public** **class** **callactivity** **extends** **Activity** {

**int** mMainValue = 0;

**TextView** mMainText;

**TextView** mBackText;

**EditText** mNumEdit;

**CalcThread** mThread;

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

**setContentView**(**R**.**layout**.*thread\_loopertest*);

mMainText = (**TextView**)**findViewById**(**R**.**id**.*mainvalue*);

mBackText = (**TextView**)**findViewById**(**R**.**id**.*backvalue*);

mNumEdit = (**EditText**)**findViewById**(**R**.**id**.*number*);

**Button** btn = (**Button**)**findViewById**(**R**.**id**.*increase*);

btn.**setOnClickListener**(**new** **Button**.**OnClickListener**() {

**public** **void** **onClick**(**View** v) {

mMainValue++;

mMainText.**setText**("MainValue : " + mMainValue);

}

});

btn = (**Button**)**findViewById**(**R**.**id**.*square*);

btn.**setOnClickListener**(**new** **Button**.**OnClickListener**() {

**public** **void** **onClick**(**View** v) {

**Message** msg = **Message**.*obtain*();

msg.what = 0;

msg.arg1 = **Integer**.*parseInt*(mNumEdit.**getText**().**toString**());

mThread.mBackHandler.**sendMessage**(msg);

}

});

btn = (**Button**)**findViewById**(**R**.**id**.*root*);

btn.**setOnClickListener**(**new** **Button**.**OnClickListener**() {

**public** **void** **onClick**(**View** v) {

**Message** msg = **Message**.*obtain*();

msg.what = 1;

msg.arg1 = **Integer**.*parseInt*(mNumEdit.**getText**().**toString**());

mThread.mBackHandler.**sendMessage**(msg);

}

});

mThread = **new** **CalcThread**(mHandler);

mThread.**setDaemon**(**true**);

mThread.**start**();

}

**Handler** mHandler = **new** **Handler**() {

**public** **void** **handleMessage**(**Message** msg) {

**switch** (msg.what) {

**case** 0:

mBackText.**setText**("Square Result : " + msg.arg1);

**break**;

**case** 1:

mBackText.**setText**("Root Result : " + ((**Double**)msg.obj).**doubleValue**());

**break**;

}

}

};

}

**class** **CalcThread** **extends** **Thread** {

루퍼를준비한다.

루프를돌면서 감시한다

메인에는 루퍼가있지만 여기에는없기 때문에 생만들어줘야한다.

**Handler** mMainHandler;

**CalcThread**(**Handler** handler) {

mMainHandler = handler;

}

**public** **void** **run**() {

**Looper**.*prepare*();

**Looper**.*loop*();

}

**public** **Handler** mBackHandler = **new** **Handler**() {

**public** **void** **handleMessage**(**Message** msg) {

**Message** retmsg = **Message**.*obtain*();

**switch** (msg.what) {

**case** 0:

**try** { **Thread**.*sleep*(100); } **catch** (**InterruptedException** e) {;}

retmsg.what = 0;

retmsg.arg1 = msg.arg1 \* msg.arg1;

**break**;

**case** 1:

**try** { **Thread**.*sleep*(100); } **catch** (**InterruptedException** e) {;}

retmsg.what = 1;

retmsg.obj = **new** **Double**(**Math**.*sqrt*((**double**)msg.arg1));

**break**;

}

mMainHandler.**sendMessage**(retmsg);

} };}

|  |  |
| --- | --- |
| Static void prepare() | 현재스레드를 위한 루퍼를 준비한다  내부적으로 메시지를 저장하기 위한 메시지큐를 생성하고 기타메시지 전송에 필요한 조치를 할것이다 |
| Static void loop() | 큐에서 메시지를 꺼내 핸들러로 전달하는 루프를 샐행한다 quit메서드가 호출되어 루푸가 종료될때까지. 무한반복 |
| Static quit() | 루푸종료 |
| Thread getThread() | 루퍼와 연결된 스레드를 구한다 |
| Static Looper getMainLooper() | 스레드 에서 반대로 루퍼를 구할 때 getMainLooper myLooper 메서드를쓴다 |
| Static Looper myLooper() | 스레드 에서 반대로 루퍼를 구할 때 getMainLooper myLooper 메서드를쓴다 |

작업스케줄링

위는 쓰레드간 통신이지만 작업스케줄에도 사용가능하다

메시지큐는 선입선출의 원칙에 따라 운용되며 먼저 들어온 메시지가 우선적으로 처리되므로 큐에 작업들을 쌓아 놓으면 순차적으로 실행될것이다.

메시지큐가 작업의 목록과 순서를 관리해주는것이다. 뿐만 아니라 메시지큐에 넣을 때 시간을 지정하여 원하는 때에 메시지를 처리하도록 등록해 놓을수도 있다 이때는 다음메서드 사용

**Boolean sendMessageAtTime(Message msg , long upTimeMillis);**

**Boolean sendMessaDelayed(Message msg , long upTimeMillis);**

절대적인 시간을 지정할수도 있고 일정 시간후를 지정할수도 있다.

위내용은 메시지꺼고 런레블은 아래거다

**Boolean postAtTime(runnable r, long uptimeMillis);**

**Boolean postAtDelayed(runnable r, long uptimeMillis);**

**public** **class** **callactivity** **extends** **Activity** **implements** **View**.OnClickListener {

**Button** mUpload;

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

**setContentView**(**R**.**layout**.*thread\_upload*);

mUpload = (**Button**)**findViewById**(**R**.**id**.*upload*);

mUpload.**setOnClickListener**(**this**);

}

**public** **void** **onClick**(**View** v) {

**switch** (v.**getId**()) {

**case** **R**.**id**.*upload*:

**new** **AlertDialog**.**Builder**(**callactivity**.**this**)

.**setTitle**("질문")

.**setMessage**("업로드 하시겠습니까?")

.**setPositiveButton**("예", **new** DialogInterface.**OnClickListener**() {

**public** **void** **onClick**(DialogInterface dialog, **int** whichButton) {

**doUpload**();

}

})

.**setNegativeButton**("아니오", **null**)

.**show**();

**문제점이 있다 예를누르면 대화상자가 안닫히고 업로드를 시작한다 업로드가 끝난후 닫히고 토스트가 뜬다.   
그이유는 온클릭에 장기간 로직을 돌리는 doUpload를 하기때문이다.**

**break**;

}

}

**void** **doUpload**() {

**for** (**int** i = 0; i < 20; i++) {

**try** { **Thread**.*sleep*(100); } **catch** (**InterruptedException** e) {;}

}

**Toast**.*makeText*(**this**, "업로드를 완료했습니다.", 0).**show**();

}}

**위내용을 보강한것이다.**

**/\* 핸들러로 지연 호출**

**public** **class** **callactivity** **extends** **Activity** **implements** **View**.OnClickListener {

**Button** mUpload;

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

**setContentView**(**R**.**layout**.*thread\_upload*);

mUpload = (**Button**)**findViewById**(**R**.**id**.*upload*);

mUpload.**setOnClickListener**(**this**);

}

**public** **void** **onClick**(**View** v) {

**switch** (v.**getId**()) {

**case** **R**.**id**.*upload*:

**new** **AlertDialog**.**Builder**(**callactivity**.**this**)

.**setTitle**("질문")

.**setMessage**("업로드 하시겠습니까?")

.**setPositiveButton**("예", **new** DialogInterface.**OnClickListener**() {

**public** **void** **onClick**(DialogInterface dialog, **int** whichButton) {

mHandler.**sendEmptyMessageDelayed**(0,10);

}

})

.**setNegativeButton**("아니오", **null**)

.**show**();

**onClick이벤트가 닫을 수 있는 시간을준다 10mm 만큼**

**기달렸다가 큐에넣는다**

**break**;

}

}

**Handler** mHandler = **new** **Handler**() {

**public** **void** **handleMessage**(**Message** msg) {

**if** (msg.what == 0) {

**doUpload**();

}

}

};

**void** **doUpload**() {

**for** (**int** i = 0; i < 20; i++) {

**try** { **Thread**.*sleep*(100); } **catch** (**InterruptedException** e) {;}

}

**Toast**.*makeText*(**this**, "업로드를 완료했습니다.", 0).**show**();

}

}

**/\* 러너블 전달**

**public** **class** **callactivity** **extends** **Activity** **implements** **View**.OnClickListener {

**Button** mUpload;

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

**setContentView**(**R**.**layout**.*thread\_upload*);

mUpload = (**Button**)**findViewById**(**R**.**id**.*upload*);

mUpload.**setOnClickListener**(**this**);

}

**public** **void** **onClick**(**View** v) {

**switch** (v.**getId**()) {

**case** **R**.**id**.*upload*:

**new** **AlertDialog**.**Builder**(**callactivity**.**this**)

.**setTitle**("질문")

.**setMessage**("업로드 하시겠습니까?")

.**setPositiveButton**("예", **new** DialogInterface.**OnClickListener**() {

**public** **void** **onClick**(DialogInterface dialog, **int** whichButton) {

mHandler.**postDelayed**(**new** **Runnable**() {

**public** **void** **run**() {

**doUpload**();

}

},10);

}

})

.**setNegativeButton**("아니오", **null**)

.**show**();

**break**;

}

}

**Handler** mHandler = **new** **Handler**();

**void** **doUpload**() {

**for** (**int** i = 0; i < 20; i++) {

**try** { **Thread**.*sleep*(100); } **catch** (**InterruptedException** e) {;}

}

**Toast**.*makeText*(**this**, "업로드를 완료했습니다.", 0).**show**();

}

}

러너블 전송방식은 Handler 구현이 필요없어서 참으로 간단하다.

또한 방법은 View 가속한 스레드의 큐를 찾아 러너블을 전송하므로 핸들러가 아예없어도 상관없다

**Boolean post(Runnable action)**

**Boolean postDelayed(Runnable action , long deleyMillis);**

Post는 지연시간 없이 러너블을 큐에 삽입하면

postDeleyed는 지연시간을 두고 삽입함 아래코드에서 구현해보자.

**//\* 뷰의 postDelayed 호출 핸들러가 아에 없어도 상관없다**

**public** **class** **callactivity** **extends** **Activity** **implements** **View**.OnClickListener {

**Button** mUpload;

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

**setContentView**(**R**.**layout**.*thread\_upload*);

mUpload = (**Button**)**findViewById**(**R**.**id**.*upload*);

mUpload.**setOnClickListener**(**this**);

}

**public** **void** **onClick**(**View** v) {

**switch** (v.**getId**()) {

**case** **R**.**id**.*upload*:

**new** **AlertDialog**.**Builder**(**callactivity**.**this**)

.**setTitle**("질문")

.**setMessage**("업로드 하시겠습니까?")

.**setPositiveButton**("예", **new** DialogInterface.**OnClickListener**() {

**public** **void** **onClick**(DialogInterface dialog, **int** whichButton) {

mUpload.**postDelayed**(**new** **Runnable**() {

**public** **void** **run**() {

**doUpload**();

}

},10);

}

})

.**setNegativeButton**("아니오", **null**)

.**show**();

**break**;

}

}

**void** **doUpload**() {

**for** (**int** i = 0; i < 20; i++) {

**try** { **Thread**.*sleep*(100); } **catch** (**InterruptedException** e) {;}

}

**Toast**.*makeText*(**this**, "업로드를 완료했습니다.", 0).**show**();

}

}

**ANR**

쓰레드활용 빠르면장땡

응용프로그램이5초이상 브로드캐스트리시버 (BR) 10 내로리턴되지않으면 죽을수도있다

**public** **class** **callactivity** **extends** **Activity** **implements** **View**.OnClickListener {

**Button** mUpload;

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

**setContentView**(**R**.**layout**.*thread\_upload*);

mUpload = (**Button**)**findViewById**(**R**.**id**.*upload*);

mUpload.**setOnClickListener**(**this**);

}

**public** **void** **onClick**(**View** v) {

**switch** (v.**getId**()) {

**case** **R**.**id**.*upload*:

**doUpload**();

**Toast**.*makeText*(**this**, "업로드를 완료했습니다.", 0).**show**();

**break**;

}

}

**void** **doUpload**() {

**for** (**int** i = 0; i < 100; i++) {

**try** { **Thread**.*sleep*(100); } **catch** (**InterruptedException** e) {;}

}

}

}

이렇게 10초이상 기다리는상태에서 이벤트가 일어나면 5초후 이벤트가일어나면

뜬다

**쓰레드로 백그라운드로 돌리자 (오래걸리는건)**

**public** **class** **callactivity** **extends** **Activity** **implements** **View**.OnClickListener {

**Button** mUpload;

**boolean** bUploading = **false**;

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

**setContentView**(**R**.**layout**.*thread\_upload*);

mUpload = (**Button**)**findViewById**(**R**.**id**.*upload*);

mUpload.**setOnClickListener**(**this**);

}

**public** **void** **onClick**(**View** v) {

**switch** (v.**getId**()) {

**case** **R**.**id**.*upload*:

**if** (bUploading) **return**;

**Thread** uploadThread = **new** **Thread**() {

**public** **void** **run**() {

**doUpload**();

mCompleteHandler.**sendEmptyMessage**(0);

}

};

bUploading = **true**;

uploadThread.**start**();

**break**;

}

}

**public** **Handler** mCompleteHandler = **new** **Handler**() {

**public** **void** **handleMessage**(**Message** msg) {

bUploading = **false**;

**Toast**.*makeText*(**callactivity**.**this**, "업로드를 완료했습니다.", 0).**show**();

}

};

**void** **doUpload**() {

**for** (**int** i = 0; i < 100; i++) {

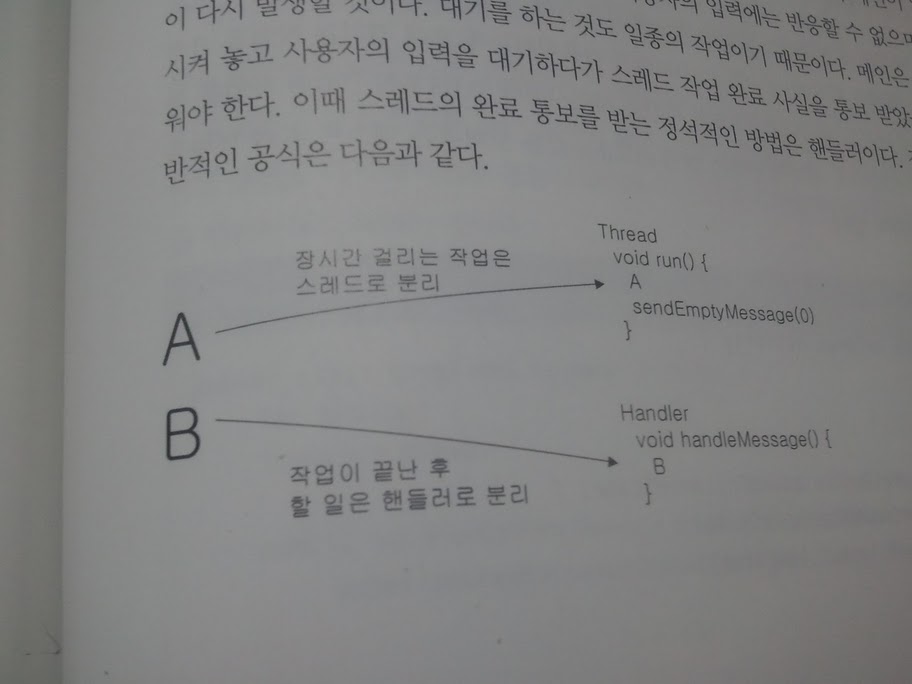
**try** { **Thread**.*sleep*(100); } **catch** (**InterruptedException** e) {;}

} }}

**쓰레드 생성즉시 리턴 작업거리를 스레드에 넘겨주고 메인스레드 계속 사용자 입력 처리한다.**

**10초후에 토스트가뜬다 ANR은 발생안한다**

쓰레드에서는 UI관련된 어떠한 동작도 할수 없으며 토스트를 출력하는것도 허용안한다



옆에 와같이 하는것이

이상적인 방법이다

DB셀렉팅같은거 플러그새워서 매번안하게 해라..

**LogTime**

복잡한연산/긴xml파싱/DB작업/네트워크입출력 처럼 본질적으로 오래걸리는작업들.. 처리.

일반적인 타이머 방법

**public** **class** **callactivity** **extends** **Activity** {

**int** mValue;

**TextView** mText;

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

**setContentView**(**R**.**layout**.*thread\_longtime*);

mText=(**TextView**)**findViewById**(**R**.**id**.*text*);

**Button** btnUpdate = (**Button**)**findViewById**(**R**.**id**.*update*);

btnUpdate.**setOnClickListener**(**new** **Button**.**OnClickListener**() {

**public** **void** **onClick**(**View** v) {

mValue = 0;

mHandler.**sendEmptyMessage**(0);

}

});

}

**Handler** mHandler = **new** **Handler**() {

**public** **void** **handleMessage**(**Message** msg) {

mValue++;

mText.**setText**(**Integer**.*toString*(mValue));

**try** { **Thread**.*sleep*(50); } **catch** (**InterruptedException** e) {;}

**if** (mValue < 100) {

mHandler.**sendEmptyMessage**(0);

}

}

};

}

**이렇게 쓰레드로 구현을 하면 오류가난다**

class UpdateThread extends Thread {

public void run() {

mValue++;

mText.**setText**(**Integer**.*toString*(mValue));

**try** { **Thread**.*sleep*(50); } **catch** (**InterruptedException** e) {;}

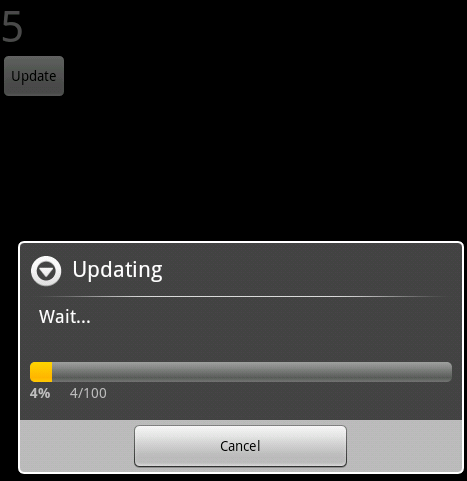
**if** (mValue < 100) {

mHandler.**sendEmptyMessage**(0);

}

}

}

**대화상자를 통해 표시**

**public** **class** **callactivity** **extends** **Activity** {

**int** mValue;

**TextView** mText;

**ProgressDialog** mProgress;

**boolean** mQuit;

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

**setContentView**(**R**.**layout**.*thread\_longtime*);

mText=(**TextView**)**findViewById**(**R**.**id**.*text*);

**Button** btnUpdate = (**Button**)**findViewById**(**R**.**id**.*update*);

btnUpdate.**setOnClickListener**(**new** **Button**.**OnClickListener**() {

**public** **void** **onClick**(**View** v) {

mValue = 0;

**showDialog**(0);

mQuit = **false**;

mHandler.**sendEmptyMessage**(0);

}

});

}

**protected** **Dialog** **onCreateDialog**(**int** id) {

**switch** (id) {

**case** 0:

mProgress = **new** **ProgressDialog**(**this**);

mProgress.**setProgressStyle**(**ProgressDialog**.*STYLE\_HORIZONTAL*);

mProgress.**setTitle**("Updating");

mProgress.**setMessage**("Wait...");

mProgress.**setCancelable**(**false**);

mProgress.**setButton**("Cancel", **new** DialogInterface.**OnClickListener**() {

**public** **void** **onClick**(DialogInterface dialog, **int** whichButton) {

mQuit = **true**;

**dismissDialog**(0);

}

});

**return** mProgress;

}

**return** **null**;

}

**Handler** mHandler = **new** **Handler**() {

**public** **void** **handleMessage**(**Message** msg) {

mValue++;

mText.**setText**(**Integer**.*toString*(mValue));

**try** { **Thread**.*sleep*(50); } **catch** (**InterruptedException** e) {;}

**if** (mValue < 100 && mQuit == **false**) {

mProgress.**setProgress**(mValue);

mHandler.**sendEmptyMessage**(0);

} **else** {

**dismissDialog**(0);

}

} }; }

**쓰레드로 분배를하자**

**public** **class** **callactivity** **extends** **Activity** {

**int** mValue;

**TextView** mText;

**ProgressDialog** mProgress;

**boolean** mQuit;

**UpdateThread** mThread;

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

**setContentView**(**R**.**layout**.*thread\_longtime*);

mText=(**TextView**)**findViewById**(**R**.**id**.*text*);

**Button** btnUpdate = (**Button**)**findViewById**(**R**.**id**.*update*);

btnUpdate.**setOnClickListener**(**new** **Button**.**OnClickListener**() {

**public** **void** **onClick**(**View** v) {

mValue = 0;

**showDialog**(0);

mQuit = **false**;

mThread = **new** **UpdateThread**();

mThread.**start**();

}

});

}

**protected** **Dialog** **onCreateDialog**(**int** id) {

**switch** (id) {

**case** 0:

mProgress = **new** **ProgressDialog**(**this**);

mProgress.**setProgressStyle**(**ProgressDialog**.*STYLE\_HORIZONTAL*);

mProgress.**setTitle**("Updating");

mProgress.**setMessage**("Wait...");

mProgress.**setCancelable**(**false**);

mProgress.**setButton**("Cancel", **new** DialogInterface.**OnClickListener**() {

**public** **void** **onClick**(DialogInterface dialog, **int** whichButton) {

mQuit = **true**;

**dismissDialog**(0);

}

});

**return** mProgress;

}

**return** **null**;

}

**Handler** mHandler = **new** **Handler**() {

**public** **void** **handleMessage**(**Message** msg) {

mValue = msg.arg1;

mText.**setText**(**Integer**.*toString*(mValue));

**if** (mValue < 100) {

mProgress.**setProgress**(mValue);

} **else** {

mQuit = **true**;

**dismissDialog**(0);

}

}

};

**class** **UpdateThread** **extends** **Thread** {

**public** **void** **run**() {

**while** (mQuit == **false**) {

mValue++;

**Message** msg = mHandler.**obtainMessage**();

msg.arg1 = mValue;

mHandler.**sendMessage**(msg);

**try** { **Thread**.*sleep*(50); } **catch** (**InterruptedException** e) {;}

}

}

}

}

**AsyncTask (추상클래스)**

백그라운드 작업을 하기위해서 스레드 핸들러등을 각각 만들어야 하고 작업중에 핸들러를 주기적으로 호출해야하는 번거로움이 있는데 이 작업들을 대신 수행해 주는 도움이 클래스가 새로 추가되었다

**내부적으로 작업 스레드를 생성하며 필요할때마다 UI스레드에서 실행되는 콜백 메서드를 호출한다 콜백메서드에서 정해진 작업만 처리하면 간단하게 백그라운드 작업을 수행할수있다.**

추상클래스이기 때문에 구현한다음에 콜백메서드를 호출한다

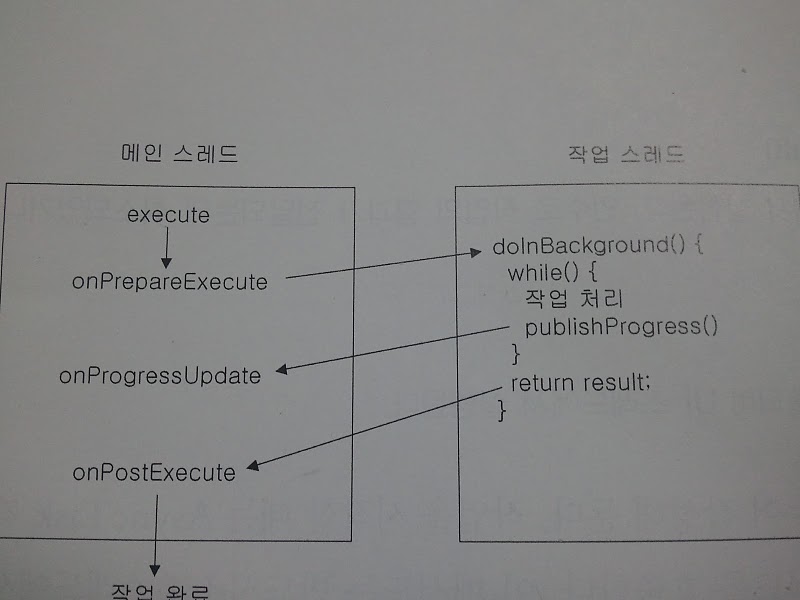
* Params 실행시에 전달할 인수의 타입이다. 즉 배경 작업거리이다.
* Progress 매작업 단계마다 진행 상태를 표기하기 위한 타입이다.
* Result 작업의 결과로 리턴될 타입이다.

각 작업단계에서 호출되는 메서드

|  |  |
| --- | --- |
| Void onPreExecute() | 작업이 시작되기전에 호출되며 UI스레드에서 실행  초기화나 프로그래스 대화상자를 준비하는 등 작업수행 |
| Result doInBackground(Params…) | 배경 작업을 수행하며 분리된 스레드에서 실행된다 execute메서드로 전달한 작업거리가 params인수로 전달되는데 여러개의 인수를 전달할수 있으므로 배열 타입으로 되어있다 하나으 인수만 필요하다면 paramsp[0] 만사용 작업중  publishProgress 메서드를 호출하여 작업 경과를 UI스레드로 보고할수 있다 최정적으로 작업된 결과를 Result타입리턴 |
| Void onProgressUpdate(Progress…) | doInBackground 에서 publishProgress 메서드를 호출할 때 작업 경과 표시를 위해 호출되며 UI스레드에서 실행된다 프로그래스 바에 진행 상태를 표시하는 역할을 한다 얼마나 자주 호출될것인가는 정의되어있지 않으므로 매 경과마다 호출된다고 보장할수없다 |
| Void onPostExecute(Result result) | 백그라운드 작업이 끝난후 UI 스레드에서 실행된다 인수로 작업의 결과가 전달되는데 취소되었거나 예외가 발생시 NULL |
| Void onCancelled() | cancel메서드로 작업을 취소했을 때 호출 UI에서 실행된다 |
| Boolean cancel(Boolean myInterruptifRunning)  Boolean isCancelled() | 작업을 취소하라고 지시한다 작업 취소가 항상 성공적이지는 않으며 실패할수도있다 이미 완려되었거나 기타 이유로 취소할수 없는 경우도 있다  isCanceled 메서드는 정상 종료되지 않고 취소되었는지 확인한다 |
| Result get(…) | 작업이 완료되기까지 대기하며 작업 결과를 돌려 받는다 필요할 경우 대기할 타임아웃값을 지정할수 있다. |
| ASYNCTASK.Status getStatus() | 작업의 현재 상태를 조사한다 아직 시작하지않은 상태라면 PENDING이 리턴되며 이미 실행중이면 RUNNING 작업이 완료면 FINISHED 리턴 |

전체적인 메서드 실행 흐름은 다음과같다

* 메인 스레드와 작업 스레들알아서 스위칭하면서 백그라운드 작업과 UI 갱신을 순서대로 실행하는 구조



AsyncTask 의 콜백 메서드로 정해져 있다는점만 다르다 doInBackground 메서드는 작업 스레드에 해당하며 onProgressUpdate는 핸들러에 해당하는 셈이다

구현해보자 **AsyncTask**

**public** **class** **callactivity** **extends** **Activity** {

**int** mValue;

**TextView** mText;

**ProgressDialog** mProgress;

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

**setContentView**(**R**.**layout**.*thread\_longtime*);

mText = (**TextView**)**findViewById**(**R**.**id**.*text*);

**Button** btnUpdate = (**Button**)**findViewById**(**R**.**id**.*update*);

btnUpdate.**setOnClickListener**(**new** **Button**.**OnClickListener**() {

**public** **void** **onClick**(**View** v) {

**new** **AccumulateTask**().**execute**(100);

}

});

}

**class** **AccumulateTask** **extends** **AsyncTask**<**Integer**, **Integer**, **Integer**> {

**protected** **void** **onPreExecute**() {

mValue = 0;

mProgress = **new** **ProgressDialog**(**callactivity**.**this**);

mProgress.**setProgressStyle**(**ProgressDialog**.*STYLE\_HORIZONTAL*);

mProgress.**setTitle**("Updating");

mProgress.**setMessage**("Wait...");

mProgress.**setCancelable**(**false**);

mProgress.**setProgress**(0);

mProgress.**setButton**("Cancel", **new** DialogInterface.**OnClickListener**() {

**public** **void** **onClick**(DialogInterface dialog, **int** whichButton) {

**cancel**(**true**);

}

});

mProgress.**show**();

자체Thread

}

**protected** **Integer** **doInBackground**(**Integer**... arg0) {

**while** (**isCancelled**() == **false**) {

mValue++;

**if** (mValue <= 100) {

**publishProgress**(mValue);

} **else** {

**break**;

}

**try** { **Thread**.*sleep*(50); } **catch** (**InterruptedException** e) {;}

}

**return** mValue;

}

**protected** **void** **onProgressUpdate**(**Integer**... progress) {

mProgress.**setProgress**(progress[0]);

mText.**setText**(**Integer**.*toString*(progress[0]));

}

**protected** **void** **onPostExecute**(**Integer** result) {

백그라운드 작업끝난뒤 실행

mProgress.**dismiss**();

}

**protected** **void** **onCancelled**() {

mProgress.**dismiss**();

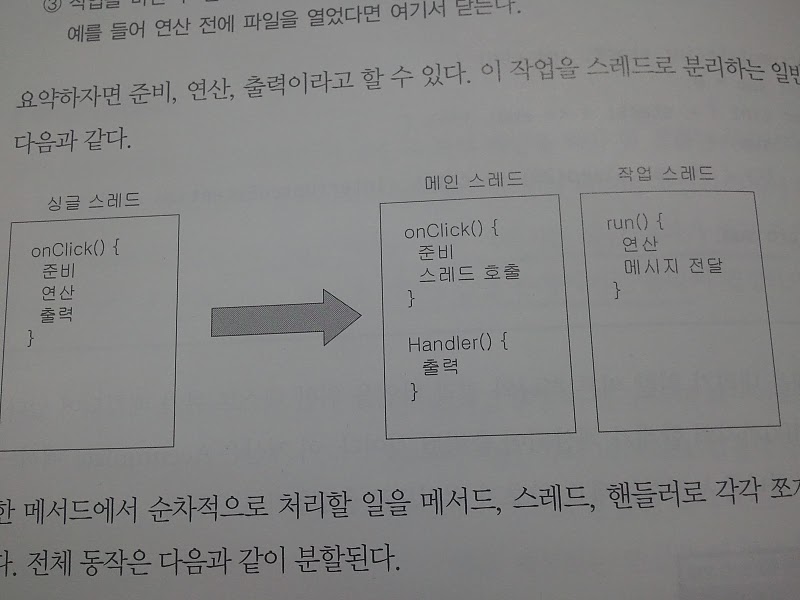
백그라운드 작업취소시. UI스레드에서 호출

}

}

**BACKWORK 674…**

중간규모 이상 프로그램 동시에 두가지 작업을 해야할경우가 많다. 쓰레드로 처리하자.



**public** **class** **callactivity** **extends** **Activity** {

**TextView** mResult;

**AccumThread** mThread;

**ProgressDialog** mProgress;

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

**setContentView**(**R**.**layout**.*thread\_backwork*);

mResult = (**TextView**)**findViewById**(**R**.**id**.*result*);

**Button** btn = (**Button**)**findViewById**(**R**.**id**.*start*);

btn.**setOnClickListener**(**new** **Button**.**OnClickListener**() {

**public** **void** **onClick**(**View** v) {

**int** start = 0, end = 100;

mProgress = **new** **ProgressDialog**(v.**getContext**());

mProgress.**setProgressStyle**(**ProgressDialog**.*STYLE\_SPINNER*);

mProgress.**setTitle**("Calculating");

mProgress.**setMessage**("Wait...");

mProgress.**setCancelable**(**false**);

mProgress.**setButton**("Cancel", **new** DialogInterface.**OnClickListener**() {

**public** **void** **onClick**(DialogInterface dialog, **int** whichButton) {

mThread.mQuit = **true**;

mProgress.**dismiss**();

}

});

mProgress.**show**();

mThread = **new** **AccumThread**(start, end, mAfterAccum);

mThread.**start**();

}

});

}

**class** **AccumThread** **extends** **Thread** {

**int** mStart, mEnd;

**int** mResult;

**Handler** mAfter;

**boolean** mQuit;

**AccumThread**(**int** start, **int** end, **Handler** after) {

mStart = start;

mEnd = end;

mQuit = **false**;

mAfter = after;

}

**public** **void** **run**() {

mResult = 0;

**for** (**int** i = mStart; i <= mEnd; i++) {

mResult += i;

**try** { **Thread**.*sleep*(20); } **catch** (**InterruptedException** e) {;}

**if** (mQuit) {

**return**;

}

}

mAfter.**sendEmptyMessage**(0);

}

}

**Handler** mAfterAccum = **new** **Handler**() {

**public** **void** **handleMessage**(**Message** msg) {

mProgress.**dismiss**();

**if** (mThread.mQuit == **false**) {

mResult.**setText**("" + mThread.mResult);

}

}

};

}

## [안드로이드 021: Thread 구현하기 2/2 (with AsyncTask & ProgressBar)](http://tigerwoods.tistory.com/28) <http://tigerwoods.tistory.com/28>

|  |  |
| --- | --- |
|  |  |

[안드로이드 Study 목차가기](http://tigerwoods.tistory.com/20)

예제 프로젝트 다운 받기

[[http://cfs.tistory.com/blog/image/extension/zip.gif](http://tigerwoods.tistory.com/attachment/cfile25.uf@1350220D4B988EC90CEC3F.zip) 021\_Thread02\_AsyncTask.zip](http://tigerwoods.tistory.com/attachment/cfile25.uf@1350220D4B988EC90CEC3F.zip)

[전 포스트](http://tigerwoods.tistory.com/26)에서 설명했던 여러 스레드 구현방법들은 비록 아무 문제가 없지만 구현방법이 복잡해서 코드를 읽기 힘들게 만드는 경향이 있었다. Background작업에 관한 모든 사항(스레드 객체 생성, 사용, UI스레드와 통신 등)이 Activity 코드에 포함 되고 특히 background 스레드가 UI위젯과 빈번한 통신을 할수록 Activity 코드의 복잡함은 점점 배가 된다. 

안드로이드에서는 이런 문제를 해결하기 위해 API level 3 (1.5 version) 부터 **AsyncTask**라는 클래스를 제공하고 있다. 

AsyncTask클래스는 background작업을 위한 모든 일(스레드생성, 작업실행, UI와 통신 등)을 추상화 함으로 각각의 background작업을 객체 단위로 구현/관리 할 수 있게 하는것이 목적이다. 그림으로 표현하면 다음과 같다.

|  |
| --- |
| http://cfile6.uf.tistory.com/image/131B56204B9997C872FBA9 |

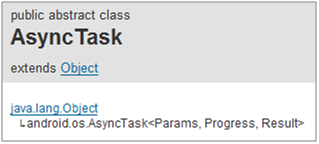
참고로 1.0과 1.1 version의 API를 사용하는 디바이스에서는 구글 code에 공개되어 있는 [UserTask](http://code.google.com/p/shelves/source/browse/trunk/Shelves/src/org/curiouscreature/android/shelves/util/UserTask.java) 라는 클래스를 어플리케이션 프로젝트에 복사해 넣어 사용할 수 있다. 기능과 사용법은 AsyncTask와 완전히 동일하다.

그럼 AsyncTask에 관해 자세히 살펴보자.

**1. AsyncTask 클래스 소개**

AsyncTask라는 클래스 이름은 Asynchronous Task의 줄임이며, UI스레드의 입장에서 볼 때 비동기적으로 작업이 수행되기 때문에 붙여진 이름이다. (Ajax: Asynchronous javascript and XML 에서 사용된 의미와 같다)

AsyncTask의 상속관계는 다음과 같다.



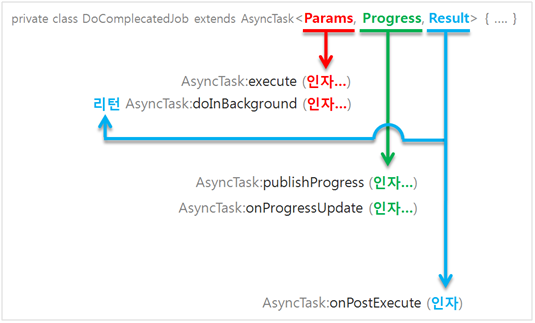
Object로부터 상속하는 AsyncTask는 Generic Class이기 때문에 사용하고자 하는 type을 지정해야 한다.

AsyncTask클래스의 사용시 지정해야 하는 generic type은 각각 다음의 용도로 사용된다.

* **Params:** background작업 시 필요한 data의 type 지정
* **Progress:** background 작업 중 진행상황을 표현하는데 사용되는 data를 위한 type 지정
* **Result:** background 작업 완료 후 리턴 할 data 의 type 지정

그림으로 각 generic type이 결정하는 것들을 표현하면 다음과 같다.

(각 메소드의 자세한 정보는 다음 단락의 예제 코드와 설명 참조)



만약 type을 정할 필요가 없는 generic이 있다면 **void**를 전달하면 된다.

예. …AsyncTask<void, void, void> {…}

**2. AsyncTask의 사용**

우선 AsyncTask가 어떻게 사용되는지 예제 소스를 보자.

**AsyncTask 클래스의 사용 예**

소스 펼치기

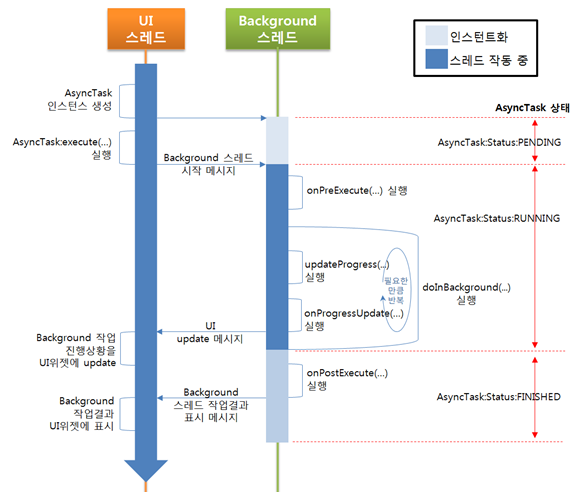
AsyncTask 클래스는 다음과 같이 중요한 callback들을 제공 함으로 상황에 맞게 오버라이딩 해야 한다.

* **protected void onPreExecute():** Background 작업이 시작되자마자 UI스레드에서 실행될 코드를 구현해야 함. (예. background 작업의 시작을 알리는 text표현, background 작업을 위한 ProgressBar popup등)
* **protected abstract Result doInBackground(Params… params):**Background에서 수행할 작업을 구현해야 함. execute(…) 메소드에 입력된 인자들을 전달 받음.
* **void onProgressUpdate(Progress... values):**publishProgress(…) 메소드 호출의 callback으로 UI스레드에서 보여지는 background 작업 진행 상황을 update하도록 구현함. (예. ProgressBar 증가 등)
* **void onPostExecute(Result result):** doInBackground(…)가 리턴하는 값을 바탕으로 UI스레드에 background 작업 결과를 표현하도록 구현 함. (예. background작업을 계산한 복잡한 산술식에 대한 답을 UI 위젯에 표현함 등)
* **void onCancelled():** AsyncTask:cancel(Boolean) 메소드를 사용해 AsyncTask인스턴스의 background작업을 정지 또는 실행금지 시켰을 때 실행되는 callback. background작업의 실행정지에 따른 리소스복구/정리 등이 구현될 수 있다.

또, AsyncTask 클래스는 background 작업의 시작과 background 작업 중 진행정보의 UI스레드 표현을 위해 다음과 같은 메소드를 제공한다.

* **final AsyncTask<…> execute(Params… params):** Background 작업을 시작한다. 꼭 UI스레드에서 호출하여야 함. 가변인자를 받아들임으로 임의의 개수의 인자를 전달할 수 있으며, 인자들은 doInBackground(…) 메소드로 전달된다.
* **final void publishProgress(Progress... values):**Background 작업 수행 중 작업의 진행도를 UI 스레드에 전달 함. doInBackground(…)메소드 내부에서만 호출.

위의 메소드들은 AsyncTask 클래스를 이용해 구현된 background 작업 시 다음과 같은 형태로 사용된다.



위 의 그림에서 처럼AsyncTask인스턴스는 자기 자신을 pending, running, finished 이렇게 세 가지 상태(status)로 구분하는데 각각 AsyncTask:Status 클래스에 상수 PENDING, RUNNING, FINISHED로 표현 될 수 있다.

현재 AsyncTask인스턴스의 상태는 다음 메소드를 호출해서 얻을 수 있다.

**public final AsyncTask.Status getStatus ()**

**return**

AsyncTask인스턴스의 상태정보를 AsyncTask.Status 객체의 상수 값 PENDING, RUNNING, FINISHED 중에서 리턴.

또, AsyncTask클래스는 background 작업을 정지, 또는 시작금지 시키기 위해 다음 메소드를 제공한다. 이 메소드가 성공적으로 호출되면 onCacelled() callback이 호출되니 onCacelled()에 적절한 뒤처리를 해주어야 한다.

final boolean cancel (boolean mayInterruptIfRunning)

**parameter**

**mayInterruptIfRunning:** true값을 제공했을 때 background작업이 실행 중일 경우(running 상태) 작업을 중단 시키고, 준비 중(pending 상태) 일 경우 작업을 실행 금지 시킴. (execute() 명령 사용 불가. 사용하면 exception 발생)

**return**

**true:** background작업을 성공적으로 중지하거나 실행 금지 시킴

**false:** 벌써 작업이 완료된 상태(finished 상태) 일 경우 리턴

마지막으로 AsyncTask 사용해 background작업을 구현 시 꼭 지켜야 하는 사항들이다.

* AsyncTask클래스는 항상 subclassing 하여 사용하여야 한다.
* AsyncTask 인스턴스는 항상 UI 스레드에서 생성한다.
* AsyncTask:execute(…) 메소드는 항상 UI 스레드에서 호출한다.
* AsyncTask:execute(…) 메소드는 생성된 AsyncTask 인스턴스 별로 꼭 한번만 사용 가능하다. 같은 인스턴스가 또 execute(…)를 실행하면 exception이 발생하며, 이는 AsyncTask:cancel(…) 메소드에 의해 작업완료 되기 전 취소된 AsyncTask 인스턴스라도 마찬가지이다. 그럼으로 background 작업이 필요할 때마다 new 연산자를 이용해 해당 작업에 대한 AsyncTask 인스턴스를 새로 생성해야 한다.
* AsyncTask의 callback 함수 onPreExecute(), doInBackground(…), onProgressUpdate(…), onPostExecute(…)는 직접 호출 하면 안 된다. (꼭 callback으로만 사용)

[안드로이드 Study 목차가기](http://tigerwoods.tistory.com/20)

**인터넷**

getSystemService(CONNECTIVITY\_SERVICE);

로 현재 가능한 통신 가져온다

CONNECTIVITY\_SERVICE는 연결관리자이다.

* 사용가능한 네트워크 정보조사
* 각 연결 방법의 현재 상태 조사
* 네트워크 연결 상태 변경될 때 모든 응용 프로그램 인텐트로 알림
* 한연결에 실패하면 대체연결 찾음

**<uses-permission android:name=*"android.permission.ACCESS\_NETWORK\_STATE"*></uses-permission>**

**퍼미션 추가.~!!**

**public** **class** **callactivity** **extends** **Activity** {

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

**setContentView**(**R**.**layout**.*network\_conmgr*);

**EditText** result = (**EditText**)**findViewById**(**R**.**id**.*result*);

**String** sResult = "";

**ConnectivityManager** mgr = (**ConnectivityManager**)

**getSystemService**(*CONNECTIVITY\_SERVICE*);

**NetworkInfo**[] ani = mgr.**getAllNetworkInfo**();

**for** (**NetworkInfo** n : ani) {

sResult += (n.**toString**() + "\n\n");

}

**NetworkInfo** ni = mgr.**getActiveNetworkInfo**();

sResult += ("Active : \n" + ni.**toString**() + "\n");

result.**setText**(sResult);

}

* }

지도.

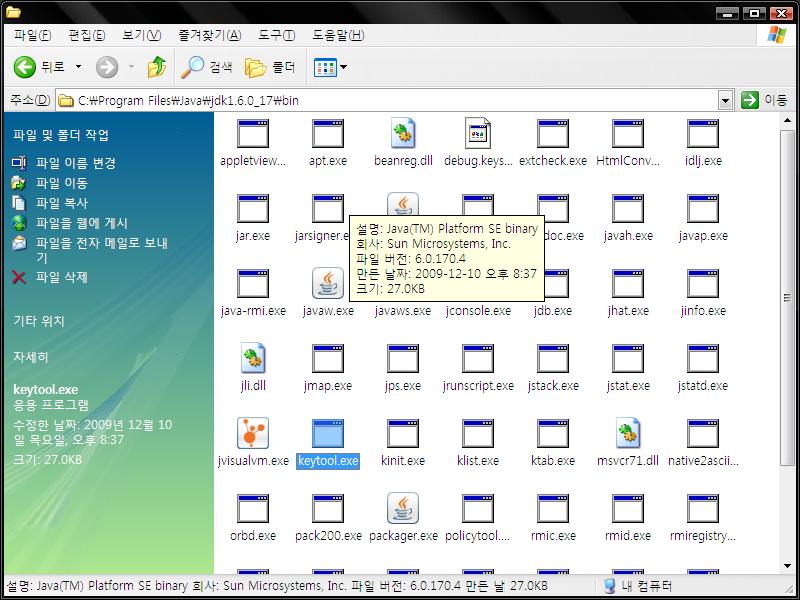
## [안드로이드에 구글맵을 사용하기 위한 준비!!](http://darkroid.textcube.com/26)

* [ANDROID?](http://darkroid.textcube.com/category/Android%3F)

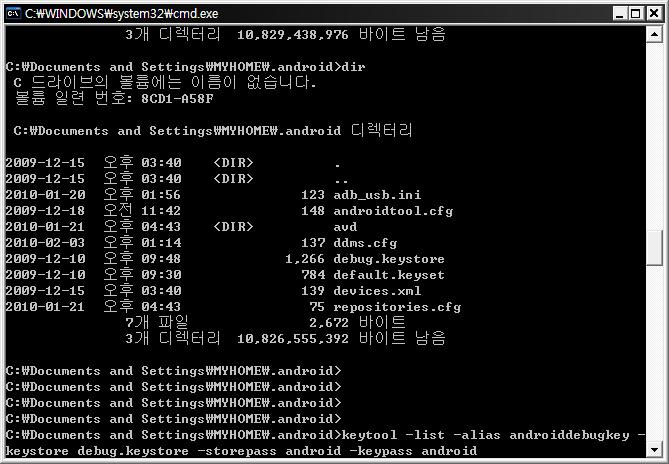
* | 2010/02/09 07:22

|  |  |
| --- | --- |
|  |  |

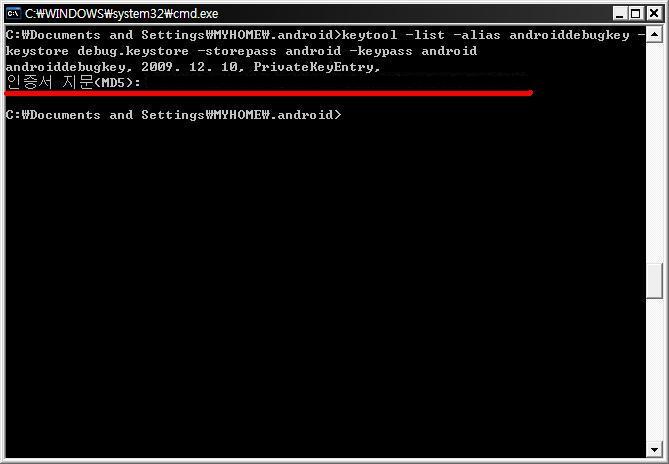
앞서 포스팅한 것들은 이미 포함되어져 있는 라이브러리를 사용해왔기에 별다른 설정 없이 사용을 해왔지만 구글맵을 사용하기 위해서는 또다른 설정을 해주어야 합니다.  
  
이 설정을 해주기 위해서는 API KEY가 필요한데 이번 포스팅에서는 이 Key를 가져오는 방법을 포스팅 하겠습니다.  
  
별로 크게 어려운 점은 없으니 차근차근 따라하시면 될 듯합니다.  
  
먼저 여러분이 미리 설치해두신 JDK가 설치된 폴더의 bin폴더에 있는 keytool을 이용해야 합니다.  
  
먼저 자신이 설치한 JDK의 설치폴더를 확인해주세요

[](http://fs.textcube.com/blog/6/67465/attach/XQisWVHwoZ.jpg)

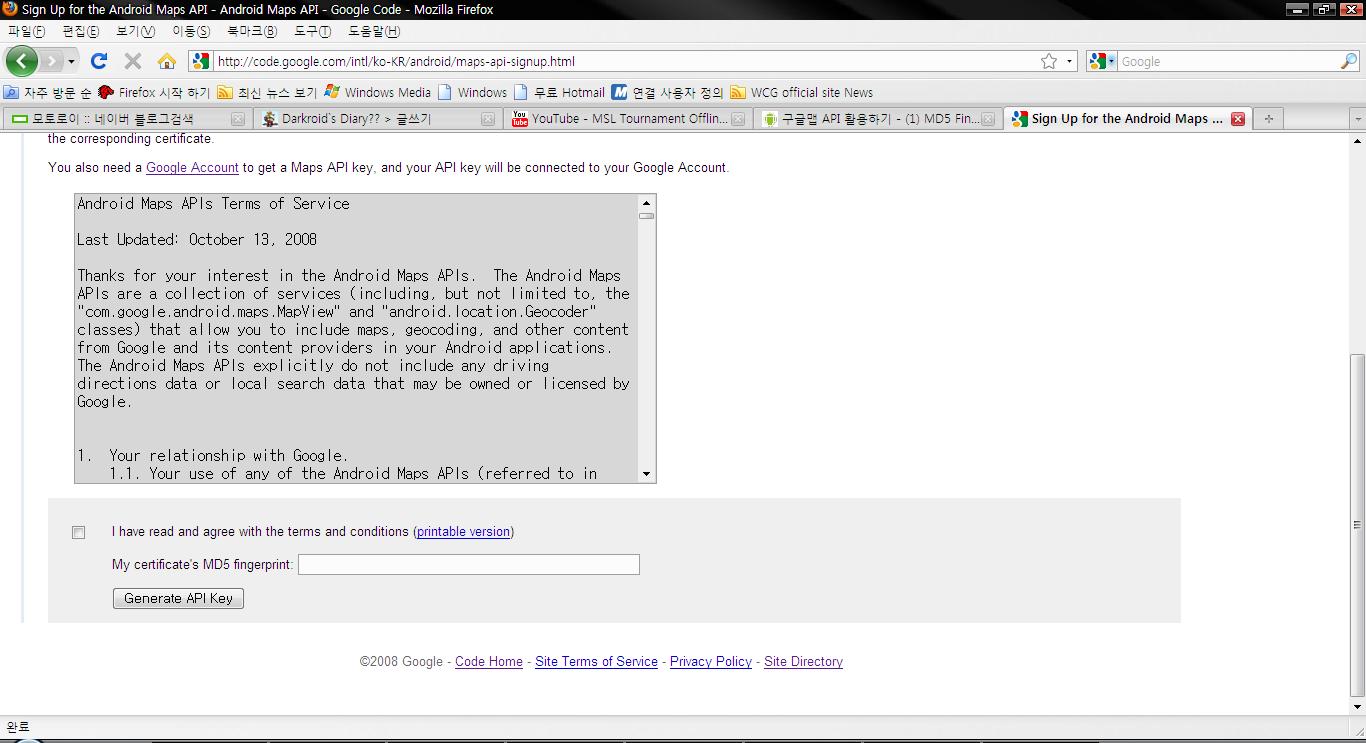
keytool을 사용하기 위해서는 커맨드프롬프트를 이용해야 합니다. 여기서 잠깐!! 커맨드프롬프트 상에서 자바명령어를 불편하지 않게 사용하기 위해서 path가 등록되어 있어야 합니다. 등록이 되어 있지 않다면 내컴퓨터의 속성에 고급탭으로 가셔서 환경변수를 클릭하시면 시스템 변수의 PATH 항목에 jdk의 bin폴더 경로를 입력해 주시면 됩니다. 물론 각 항목의 끝을 알리는 세미콜론(;)을 앞에 붙여주시고요.  
  
이제 keytool을 이용해서 MD5 Fingerprint를 생성하겠습니다. MD5 Fingerprint는 구글맵 API Key를 얻기위해 사용되므로 꼭 필요합니다. 커맨드 프롬프트 상에서 debug.keystore 가 있는 곳으로 이용해야 하는데 주로 C:\Documents and Settings\계정이름\.android 이 경로에 있으니 찾아보시기 바랍니다. 이 경로에도 없다면 파일찾기를 통해 debug.keystore 를 찾아보시는걸 권장합니다.   
  
그러면 커맨드 프롬프트 상에서 저 경로로 들어간 후에 다음과 같은 명령어를 타이핑 하시면 됩니다.

[](http://fs.textcube.com/blog/6/67465/attach/XA1raDFZIo.jpg)

keytool -list -alias androiddebugkey -keystore debug.keystore -storepass android -keypass android  
  
위의 명령어를 실행하시게 되면 다음과 같은 화면을 보실수 있습니다. 빨간줄로 밑줄 지어진 부분이 MD5 Fingerprint 입니다. 복사하셔서 보관해두세요.

[](http://fs.textcube.com/blog/6/67465/attach/XPzbUi3eZy.jpg)

그렇다면 이제 API key를 발급받아볼까요?  
  
<http://code.google.com/android/maps-api-signup.html> 로 접속하셔서 MD5 Fingerprint를 입력하시고 구글계정으로 로그인 하시면 Key를 발급받으실 수 있습니다.

[](http://fs.textcube.com/blog/6/67465/attach/XUMnPOUwX1.jpg)

발급받은 Key는 잃어버리지 않도록 잘 보관하시기 바랍니다. 다음에는 간단하게 구글맵을 안드로이드에 띄우는 것을 포스팅 하도록 하겠습니다.

**AdroidManifast.xml변경사항**

<?xml version=*"1.0"* encoding=*"utf-8"*?>

<manifest xmlns:android=*"http://schemas.android.com/apk/res/android"*

package=*"com.app.map"*

android:versionCode=*"1"*

android:versionName=*"1.0"*>

<application android:icon=*"@drawable/icon"* android:label=*"@string/app\_name"*>

<activity android:name=*".GPS"*

android:label=*"@string/app\_name"*>

<intent-filter>

<action android:name=*"android.intent.action.MAIN"* />

<category android:name=*"android.intent.category.LAUNCHER"* />

</intent-filter>

</activity>

<uses-library android:name=*"com.google.android.maps"*></uses-library>

</application>

<uses-sdk android:minSdkVersion=*"7"* />

<uses-permission android:name=*"android.permission.ACCESS\_FINE\_LOCATION"*></uses-permission>

<uses-permission android:name=*"android.permission.INTERNET"*></uses-permission>

</manifest>

**위치기반 서비스 API**

1. LOCATION\_SERVICE로 **getSystemService 호출 LocationManager 얻음**
2. getBestProvider() getAllProviders() 메서드이용 위치정보 제공자 찾음
3. criteria 객체를 설정해서 그존건들에 가장 근접한 제공자 선택
4. LocationListener 클래스구현.
5. requestLocationUpdates() 메서드를 호출된다 이시점부터 응용프로그램 위치정보 제공된다

**private** **boolean** **getmyLocation**() {

**LocationManager** locationManager;

locationManager = (**LocationManager**)**getSystemService**(**Context**.*LOCATION\_SERVICE*);

**Location** location = locationManager.**getLastKnownLocation**(**LocationManager**.*GPS\_PROVIDER*);

**boolean** sw=**false**;

**if** (location != **null**) {

lat=location.**getLatitude**();

lon=location.**getLongitude**();

sw = **true**;

} **else** {

lat = 36;

lon = 127.7;

}

**CustInfo** myInfo = **new** **CustInfo**("mylocal","위치","접속위치",lat,lon,999);

**addCustInfo**(myInfo);

**return** sw;

}

**public** **class** **GPS** **extends** **Activity** **implements** LocationListener{

**LocationManager** location = **null**;

@Override

**protected** **void** **onCreate**(**Bundle** savedInstanceState) {

**적절한 공급자 찾고**

**LocationManager**

**객체 얻어오고**

**super**.**onCreate**(savedInstanceState);

**setContentView**(**R**.**layout**.*gps*);

location = (**LocationManager**) **getSystemService**(**Context**.*LOCATION\_SERVICE*);

**final** **Button** start = (**Button**)**findViewById**(**R**.**id**.*start*);

**final** **Button** stop = (**Button**)**findViewById**(**R**.**id**.*stop*);

**final** **TextView** status = (**TextView**)**findViewById**(**R**.**id**.*status*);

start.**setOnClickListener**(**new** **View**.**OnClickListener**() {

**public** **void** **onClick**(**View** v) {

**Toast**.*makeText*(**GPS**.**this**, "Starting....", **Toast**.*LENGTH\_SHORT*).**show**();

Iterator<**String**> providers = location.**getAllProviders**().**iterator**();

**while**(providers.**hasNext**()) {

**Log**.*v*("Location", providers.**next**());

}

**Criteria** criteria = **new** **Criteria**();

criteria.**setAccuracy**(**Criteria**.*NO\_REQUIREMENT*);

criteria.**setPowerRequirement**(**Criteria**.*NO\_REQUIREMENT*);

**String** best = location.**getBestProvider**(criteria, **true**);

status.**setText**("Best provider: " + best);

location.**requestLocationUpdates**(best, 1000, 0, GPS.this);

**리스너 등록한거 탄다!**

**이시점부터 시작이다**

**4가지의 함수 오버라이딩해야함**

start.**setVisibility**(**View**.*GONE*);

stop.**setVisibility**(**View**.*VISIBLE*);

}

});

stop.**setOnClickListener**(**new** **View**.**OnClickListener**() {

2개 제공자의 활성화/비활성화 체크

1나 제공자 상태 알려주는거

1나 onLocationChaned가 바로 위치정보 응용프로그램에게 제공하는 메서드

**public** **void** **onClick**(**View** v) {

stop.**setVisibility**(**View**.*GONE*);

start.**setVisibility**(**View**.*VISIBLE*);

}

});

}

**Location** lastLocation = **null**;

**public** **void** **onLocationChanged**(**Location** location) {

**String** locInfo = **String**.*format*("Current loc = (%f, %f) @ (%f meters up)", location.**getLatitude**(), location.**getLongitude**(), location.**getAltitude**() );

**if** (lastLocation != **null**) {

**float** distance = location.**distanceTo**(lastLocation);

locInfo += **String**.*format*("\n Distance from last = %f meters", distance);

}

lastLocation = location;

**Geocoder** coder = **new** **Geocoder**(**this**);

**try** {

Iterator<**Address**> addresses = coder.**getFromLocation**(location.**getLatitude**(), location.**getLongitude**(), 3).**iterator**();

**if** (addresses != **null**) {

**while** (addresses.**hasNext**()) {

**Address** namedLoc = addresses.**next**();

**String** placeName = namedLoc.**getLocality**();

**String** featureName = namedLoc.**getFeatureName**();

**String** country = namedLoc.**getCountryName**();

**String** road = namedLoc.**getThoroughfare**();

locInfo += **String**.*format*("\n[%s][%s][%s][%s]", placeName, featureName, road, country);

**int** addIdx = namedLoc.**getMaxAddressLineIndex**();

**for** (**int** idx = 0; idx <= addIdx; idx++){

**String** addLine = namedLoc.**getAddressLine**(idx);

locInfo += **String**.*format*("\nLine %d: %s", idx, addLine);

}

}

}

} **catch** (**IOException** e) {

**Log**.*e*("GPS", "Failed to get address", e);

}

**TextView** status = (**TextView**)**findViewById**(**R**.**id**.*status*);

status.**setText**(locInfo);

**final** **String** geoURI = **String**.*format*("geo: %f,%f", location.**getLatitude**(), location.**getLongitude**());

**Button** show = (**Button**) **findViewById**(**R**.**id**.*show\_map*);

show.**setOnClickListener**(**new** **View**.**OnClickListener**() {

**public** **void** **onClick**(**View** v) {

**Intent** map = **new** **Intent**(**Intent**.*ACTION\_VIEW*, **Uri**.*parse*(geoURI));

**startActivity**(map);

}

});

show.**setVisibility**(**View**.*VISIBLE*);

}

**public** **void** **onProviderDisabled**(**String** provider) {

**Log**.*v*("GPS", "Provider disabled "+ provider);

}

**public** **void** **onProviderEnabled**(**String** provider) {

**Log**.*v*("GPS", "Provider enabled "+ provider);

}

**private** **static** **final** Map<**Integer**, **String**> *providerStatusMap* = **new** **HashMap**<**Integer**, **String**>() {

{

**put**(**LocationProvider**.*AVAILABLE*, "Available");

**put**(**LocationProvider**.*OUT\_OF\_SERVICE*, "Out of Service");

**put**(**LocationProvider**.*TEMPORARILY\_UNAVAILABLE*, "Temporarily Unavailable");

**put**(-1, "Not Reported");

}

};

**public** **void** **onStatusChanged**(**String** provider, **int** status, **Bundle** extras) {

**int** satellites = extras.**getInt**("satellites", -1);

**String** statusInfo = **String**.*format*("Provider: %s, status: %s, satellites: %d", provider, *providerStatusMap*.**get**(status), satellites);

**Log**.*v*("GPS", statusInfo);

**TextView** statusText = (**TextView**)**findViewById**(**R**.**id**.*status*);

statusText.**setText**(statusInfo);

}

}

**CAMERA 카메라**

1. SurfaceView 클래스를 확장 하여 SurfaceHolder.Callback 인터페이스 구현
2. surfaceCreated() 메서드에서 Camera인스턴스 얻는다.
3. surfaceChanged() 메서드에서 Camera.Paremeters를 이용해 관련 속성 설정후 startPreview()메서드 호출
4. 촬영 사건에 반응할 메서드를 CameraSurfaceView에 추가
5. CameraSurfaceView를 적절한 레이아웃추가
6. 사용자가 사진을 찍을수 있게 하는 위젯(버튼등)을 화면에 추가
7. 찍은 이미지의 저장을 위해 pictureCallback클래스 구현
8. Android.permission.CAMERA권한 AndroideManifast.xml 파일에 추가
9. surfaceDestroyed()메서드에서 camera 객체 해제.

**애니메이션**

|  |  |
| --- | --- |
| 프레임 애니메이션 | 주기적으로 그림을 갈아치우는 전통적인 방법이다. |
| 트윈 애니메이션 | 위치나 크기 회전 등의 효과를 일정시간 내에 수행하여 애니메이션을 진행 |

프레임 애니

Draw/numani.xml

<?xml version=*"1.0"* encoding=*"utf-8"*?>

<animation-list xmlns:android=*"http://schemas.android.com/apk/res/android"*

android:**oneshot=*"false"*>**

<item android:drawable=*"@drawable/frame1"* android:duration=*"700"* />

<item android:drawable=*"@drawable/frame2"* android:duration=*"700"* />

<item android:drawable=*"@drawable/frame3"* android:duration**=*"700"*** />

</animation-list>

애니메션 반복할건지 끝낼껀지 원샷 옵션. / 1/1000 초

백그라운드로 쓰면된다

**public** **class** **FrameAni** **extends** **Activity** {

**AnimationDrawable** mAni;

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

**setContentView**(**R**.**layout**.*draw\_frameani*);

**ImageView** img = (**ImageView**)**findViewById**(**R**.**id**.*count*);

mAni = (**AnimationDrawable**)img.**getBackground**();

**Button** btn = (**Button**)**findViewById**(**R**.**id**.*start*);

btn.**setOnClickListener**(**new** **Button**.**OnClickListener**() {

**public** **void** **onClick**(**View** v) {

mAni.**start**();

}

});

btn = (**Button**)**findViewById**(**R**.**id**.*stop*);

btn.**setOnClickListener**(**new** **Button**.**OnClickListener**() {

**public** **void** **onClick**(**View** v) {

mAni.**stop**();

}

});

}

}

트윈 애니메이션

xml파일은 res/anim 폴더에저장. 불러다쓸땐 R.anim.xxx

Animation

TranslateAnimation //좌표값 변경

ScaleAnimation //크기변경

RotateAnimation //회전각도

AlphaAnimation //투명도

AnimationSet //애니메이션조합

Static Animation AnimationUtils.loadAmimation(Context c, int id)정적메서드로 불러다사용

**public** **class** **Tween** **extends** **Activity** {

**LinearLayout** mLinear;

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

**setContentView**(**R**.**layout**.*draw\_tween*);

mLinear = (**LinearLayout**)**findViewById**(**R**.**id**.*linear*);

**findViewById**(**R**.**id**.*translate*).**setOnClickListener**(mClickListener);

**findViewById**(**R**.**id**.*rotate*).**setOnClickListener**(mClickListener);

**findViewById**(**R**.**id**.*scale*).**setOnClickListener**(mClickListener);

**findViewById**(**R**.**id**.*alpha*).**setOnClickListener**(mClickListener);

**findViewById**(**R**.**id**.*set*).**setOnClickListener**(mClickListener);

}

**Button**.OnClickListener mClickListener = **new** **View**.**OnClickListener**() {

**public** **void** **onClick**(**View** v) {

**Animation** ani = **null**;

**switch** (v.**getId**()) {

**case** **R**.**id**.*translate*:

ani = **AnimationUtils**.*loadAnimation*(**Tween**.**this**, **R**.**anim**.*translate*);

**break**;

**case** **R**.**id**.*rotate*:

ani = **AnimationUtils**.*loadAnimation*(**Tween**.**this**, **R**.**anim**.*rotate*);

**break**;

**case** **R**.**id**.*scale*:

ani = **AnimationUtils**.*loadAnimation*(**Tween**.**this**, **R**.**anim**.*scale*);

**break**;

**case** **R**.**id**.*alpha*:

ani = **AnimationUtils**.*loadAnimation*(**Tween**.**this**, **R**.**anim**.*alpha*);

**break**;

**case** **R**.**id**.*set*:

ani = **AnimationUtils**.*loadAnimation*(**Tween**.**this**, **R**.**anim**.*set*);

**break**;

}

mLinear.**startAnimation**(ani);

}

};

}

Rotate.xml

<?xml version=*"1.0"* encoding=*"utf-8"*?>

<set xmlns:android=*"http://schemas.android.com/apk/res/android"*

android:interpolator=*"@android:anim/accelerate\_interpolator"*>

<rotate

android:fromDegrees=*"0"*

android:toDegrees=*"360"*

android:pivotX=*"50%"*

android:pivotY=*"50%"*

android:duration=*"3000"*

/>

</set>

Set앨리먼트 애니메이션의 집합을 정아하는 단독 애니메이션이라도 set 앨리먼트 안에 작성된다.

|  |  |
| --- | --- |
| Accelerate\_interpolator | 처음에 느리다 점점빠르게 |
| Decelerate\_interpolator | 처음에 빠르다가. 느리게 |
| Acceleratedecelerate\_interpolator | 빨라지다가 다시 느려진다 |

Scale.xml

<?xml version=*"1.0"* encoding=*"utf-8"*?>

<set xmlns:android=*"http://schemas.android.com/apk/res/android"*

android:interpolator=*"@android:anim/accelerate\_interpolator"*>

<scale

android:fromXScale=*"1.0"*

android:toXScale=*"0.1"*

android:fromYScale=*"1.0"*

android:toYScale=*"0.1"*

android:pivotX=*"50%"*

android:pivotY=*"50%"*

android:duration=*"2000"*

/>

</set>

Alpha.xml

<?xml version=*"1.0"* encoding=*"utf-8"*?>

<set xmlns:android=*"http://schemas.android.com/apk/res/android"*

android:interpolator=*"@android:anim/accelerate\_interpolator"*>

<alpha

android:fromAlpha=*"1.0"*

android:toAlpha=*"0.0"*

android:duration=*"1000"*

android:repeatCount=*"1"*

android:repeatMode=*"reverse"*

/>

</set>

애니메이션 리스너.

|  |  |
| --- | --- |
| onAnimationStart(Animation animation) | 시작. |
| onAnimationEnd(Animation animation) | 끝 |
| onAnimationRepeat(Animation animation) | 되풀 |

**public** **class** **TweenListener** **extends** **Activity** {

**LinearLayout** mLinear;

**Button** mBtn;

**Animation** mAni1;

**Animation** mAni2;

**Animation** mAni3;

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

**setContentView**(**R**.**layout**.*draw\_tweenlistener*);

mLinear = (**LinearLayout**)**findViewById**(**R**.**id**.*linear*);

mBtn = (**Button**)**findViewById**(**R**.**id**.*start*);

mBtn.**setOnClickListener**(**new** **Button**.**OnClickListener**() {

**public** **void** **onClick**(**View** v) {

mBtn.**startAnimation**(mAni1);

}

});

mAni1 = **AnimationUtils**.*loadAnimation*(**this**, **R**.**anim**.*rotate2*);

mAni2 = **AnimationUtils**.*loadAnimation*(**this**, **R**.**anim**.*alpha2*);

mAni3 = **AnimationUtils**.*loadAnimation*(**this**, **R**.**anim**.*scale2*);

mAni1.**setAnimationListener**(**new** **AnimationListener**() {

**public** **void** **onAnimationEnd**(**Animation** animation) {

mBtn.**startAnimation**(mAni2);

}

**public** **void** **onAnimationRepeat**(**Animation** animation) {;}

**public** **void** **onAnimationStart**(**Animation** animation) {;}

});

mAni2.**setAnimationListener**(**new** **AnimationListener**() {

**public** **void** **onAnimationEnd**(**Animation** animation) {

mBtn.**startAnimation**(mAni3);

}

**public** **void** **onAnimationRepeat**(**Animation** animation) {;}

**public** **void** **onAnimationStart**(**Animation** animation) {;}

});

mAni3.**setAnimationListener**(**new** **AnimationListener**() {

**public** **void** **onAnimationEnd**(**Animation** animation) {

**Toast**.*makeText*(**TweenListener**.**this**, "Animation End", 0).**show**();

}

**public** **void** **onAnimationRepeat**(**Animation** animation) {;}

**public** **void** **onAnimationStart**(**Animation** animation) {;}

});

}

}

**그리기**

Canvas는 뷰의 그리기 표면이면 이위에서 그린다.

|  |  |  |  |
| --- | --- | --- | --- |
| drawPoint |  | drawArc | 각도만큼 원호그린다 세시방향0도 |
| drawLine |  |  |  |
| drawCicle |  |  |  |
| drawRect |  |  |  |
| drawText |  |  |  |

모든 그리기 메서드의 마지막 인수는 항상 Paint객체이다.

Paint는 색상이나 글꼴 스타일 그리기 모드 등의 정보를 지정하며 이정보대로 그려진다.

New 연산자로 생성하면 검정색에 기본폰트로 설정된게 나온다

Paint.setColor(..) 등 제공한다 옵션변경할수 있게

**public** **class** **Primitive1** **extends** **Activity** {

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

**MyView** vw = **new** **MyView**(**this**);

**setContentView**(vw);

}

**protected** **class** **MyView** **extends** **View** {

**public** **MyView**(**Context** context) {

**super**(context);

}

**public** **void** **onDraw**(**Canvas** canvas) {

**Paint** Pnt = **new** **Paint**();

canvas.**drawColor**(**Color**.*WHITE*);

canvas.**drawPoint**(10,10,Pnt);

Pnt.**setColor**(**Color**.*BLUE*);

canvas.**drawLine**(20,10,200,50,Pnt);

Pnt.**setColor**(**Color**.*RED*);

canvas.**drawCircle**(100,90,50,Pnt);

Pnt.**setColor**(0x800000ff);

canvas.**drawRect**(10,100,200,170,Pnt);

Pnt.**setColor**(**Color**.*BLACK*);

canvas.**drawText**("Canvas Text Out", 10,200,Pnt);

}

}

}

**public** **class** **Primitive2** **extends** **Activity** {

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

**MyView** vw = **new** **MyView**(**this**);

**setContentView**(vw);

}

**protected** **class** **MyView** **extends** **View** {

**public** **MyView**(**Context** context) {

**super**(context);

}

**public** **void** **onDraw**(**Canvas** canvas) {

**Paint** Pnt = **new** **Paint**();

//canvas.drawARGB(255,255,255,255);

//canvas.drawRGB(255,255,255);

//canvas.drawColor(0xffffffff);

Pnt.**setColor**(**Color**.*WHITE*);

canvas.**drawPaint**(Pnt);

**RectF** r=**new** **RectF**(10,10,100,100);

Pnt.**setColor**(**Color**.*RED*);

canvas.**drawRoundRect**(r,10,10,Pnt);

r.**set**(110,10,150,100);

canvas.**drawOval**(r,Pnt);

Pnt.**setColor**(**Color**.*MAGENTA*);

r.**set**(10,110,100,200);

canvas.**drawArc**(r,10,150,**false**,Pnt);

r.**set**(110,110,200,200);

canvas.**drawArc**(r,10,150,**true**,Pnt);

Pnt.**setColor**(**Color**.*BLUE*);

**float**[] pts={10,210,50,250,50,250,110,220};

canvas.**drawLines**(pts,Pnt);

Pnt.**setColor**(**Color**.*BLACK*);

**float**[] pts2={20,210,50,240,100,220};

canvas.**drawPoints**(pts2, Pnt);

}

}

}

Paint

setAntiAlias(Boolean a)

안티를 맥이면 좋아진다.

그리기 객체.

**public** **class** **DrawBitmap** **extends** **Activity** {

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

**MyView** vw = **new** **MyView**(**this**);

**setContentView**(vw);

}

**protected** **class** **MyView** **extends** **View** {

**public** **MyView**(**Context** context) {

**super**(context);

}

**public** **void** **onDraw**(**Canvas** canvas) {

**Paint** Pnt = **new** **Paint**();

canvas.**drawColor**(**Color**.*WHITE*);

/\* 원론적인 방법

Resources r=getResources();

BitmapDrawable bd=(BitmapDrawable)r.getDrawable(R.drawable.eighty8);

Bitmap bit=bd.getBitmap();

//\*/

// 간단한 방법

**Bitmap** bit = **BitmapFactory**.*decodeResource*(**getResources**(), **R**.**drawable**.*eighty8*);

canvas.**drawBitmap**(bit, 10, 10, **null**);

canvas.**drawBitmap**(bit, **null**, **new** **Rect**(80,10,110,58), **null**);

canvas.**drawBitmap**(bit, **new** **Rect**(30,40,58,90), **new** **Rect**(120,10,120+56,10+80), **null**);

**Bitmap** BackBit = **Bitmap**.*createBitmap*(200,100,**Bitmap**.Config.*ARGB\_8888*);

**Canvas** offscreen = **new** **Canvas**(BackBit);

offscreen.**drawColor**(**Color**.*GREEN*);

Pnt.**setColor**(**Color**.*RED*);

**for** (**int** x=0;x<200;x+=5) {

offscreen.**drawLine**(x,0,200-x,100,Pnt);

}

canvas.**drawBitmap**(BackBit,10,120,**null**);

}

}

}

텍스트 그리기.

**public** **class** **DrawText** **extends** **Activity** {

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

**MyView** vw = **new** **MyView**(**this**);

**setContentView**(vw);

}

**protected** **class** **MyView** **extends** **View** {

**public** **MyView**(**Context** context) {

**super**(context);

}

**public** **void** **onDraw**(**Canvas** canvas) {

**Paint** Pnt = **new** **Paint**();

canvas.**drawColor**(**Color**.*WHITE*);

**String** str = "Made in Korea";

**char**[] arCh = {'a', 'b', 'c' };

// 기본 문자열 출력. 안티 알리아싱을 적용했다.

Pnt.**setAntiAlias**(**true**);

Pnt.**setColor**(**Color**.*BLACK*);

canvas.**drawText**(str, 10, 10, Pnt);

canvas.**drawText**(str, 2, 11, 10, 30, Pnt);

canvas.**drawText**(arCh, 0, 3, 10, 50, Pnt);

// 수평 정렬

Pnt.**setTextAlign**(**Paint**.Align.*LEFT*);

canvas.**drawText**("Left Align", 100, 90, Pnt);

Pnt.**setTextAlign**(**Paint**.Align.*CENTER*);

canvas.**drawText**("Center Align", 100, 110, Pnt);

Pnt.**setTextAlign**(**Paint**.Align.*RIGHT*);

canvas.**drawText**("Right Align", 100, 130, Pnt);

// 수직 정렬은 항상 글꼴의 베이스에 맞춰진다.

Pnt.**setTextAlign**(**Paint**.Align.*LEFT*);

Pnt.**setColor**(**Color**.*RED*);

canvas.**drawText**("Small", 10, 180, Pnt);

Pnt.**setTextSize**(20);

canvas.**drawText**("Mid", 40, 180, Pnt);

Pnt.**setTextSize**(30);

canvas.**drawText**("Big", 80, 180, Pnt);

Pnt.**setTextSize**(40);

canvas.**drawText**("Hy", 125, 180, Pnt);

// 여러 가지 속성 동시에 적용해 보기

Pnt.**setColor**(**Color**.*BLUE*);

Pnt.**setTextAlign**(**Paint**.Align.*LEFT*);

Pnt.**setTextSize**(30);

Pnt.**setTypeface**(**Typeface**.*create*((**String**)**null**, **Typeface**.*BOLD*));

Pnt.**setUnderlineText**(**true**);

Pnt.**setStrikeThruText**(**true**);

Pnt.**setSubpixelText**(**true**);

Pnt.**setTextSkewX**(-0.25f);

Pnt.**setTextScaleX**(0.8f);

canvas.**drawText**(str, 10, 220, Pnt);

}

}

}

토스트 toast

**public** **class** **ToastTest** **extends** **Activity** {

**Toast** mToast = **null**;

**int** count;

**String** str;

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

**setContentView**(**R**.**layout**.*output\_toasttest*);

**findViewById**(**R**.**id**.*shortmsg*).**setOnClickListener**(mClickListener);

**findViewById**(**R**.**id**.*longmsg*).**setOnClickListener**(mClickListener);

**findViewById**(**R**.**id**.*count1*).**setOnClickListener**(mClickListener);

**findViewById**(**R**.**id**.*count2*).**setOnClickListener**(mClickListener);

**findViewById**(**R**.**id**.*customview*).**setOnClickListener**(mClickListener);

}

**Button**.OnClickListener mClickListener = **new** **Button**.**OnClickListener**() {

**public** **void** **onClick**(**View** v) {

**switch** (v.**getId**()) {

**case** **R**.**id**.*shortmsg*:

**Toast**.*makeText*(**ToastTest**.**this**, "잠시 나타나는 메시지",

**Toast**.*LENGTH\_SHORT*).**show**();

**break**;

**case** **R**.**id**.*longmsg*:

**Toast**.*makeText*(**ToastTest**.**this**, "조금 길게 나타나는 메시지",

**Toast**.*LENGTH\_LONG*).**show**();

**break**;

**case** **R**.**id**.*count1*:

str = "현재 카운트 = " + count++;

**if** (mToast != **null**) {

mToast.**cancel**();

}

mToast = **Toast**.*makeText*(**ToastTest**.**this**, str, **Toast**.*LENGTH\_SHORT*);

mToast.**show**();

**break**;

**case** **R**.**id**.*count2*:

str = "현재 카운트 = " + count++;

**if** (mToast == **null**) {

mToast = **Toast**.*makeText*(**ToastTest**.**this**, str, **Toast**.*LENGTH\_SHORT*);

} **else** {

mToast.**setText**(str);

}

mToast.**show**();

**break**;

**case** **R**.**id**.*customview*:

**LinearLayout** linear = (**LinearLayout**)**View**.*inflate*(**ToastTest**.**this**,

**R**.**layout**.*output\_toast*, **null**);

**Toast** t2 = **new** **Toast**(**ToastTest**.**this**);

t2.**setView**(linear);

t2.**show**();

**break**;

}

}

};

}

**그리기 고급**

Shader는도형의 내부 표면을 채우는 무늬이다. 밋밋한 단일(SolidColor)만 사용했는데 세이더 활용하자

Shader

LinearGradient

RadialGradient

SweepGradient

BitmapShader

ComposeShader

페인터에 쉐이더를 넣어사용한다.

Pint.setShader(Shader shader)

|  |  |
| --- | --- |
| CLAMP | 무늬 부분 계속 반복 |
| MIRROP | 무늬를 반사시켜 계속반복 |
| REPEAT | 똑같은 무늬를 계속반복 |

**public** **class** **LinearGrad** **extends** **Activity** {

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

**setContentView**(**new** **MyView**(**this**));

}

**protected** **class** **MyView** **extends** **View** {

**public** **MyView**(**Context** context) {

**super**(context);

}

**public** **void** **onDraw**(**Canvas** canvas) {

**Paint** Pnt = **new** **Paint**();

Pnt.**setAntiAlias**(**true**);

**int**[] colors = { **Color**.*RED*, **Color**.*GREEN*, **Color**.*BLUE*, **Color**.*YELLOW*, **Color**.*WHITE* };

**float**[] pos = { 0.0f, 0.1f, 0.6f, 0.9f, 1.0f };

// 수평

Pnt.**setShader**(**new** **LinearGradient**(0,0,100,0,

**Color**.*BLUE*, **Color**.*WHITE*, TileMode.*CLAMP*));

canvas.**drawRect**(0,0,100,100,Pnt);

// 우하향

Pnt.**setShader**(**new** **LinearGradient**(110,0,210,100,

**Color**.*BLUE*, **Color**.*WHITE*, TileMode.*CLAMP*));

canvas.**drawRect**(110,0,210,100,Pnt);

// 우상향

Pnt.**setShader**(**new** **LinearGradient**(220,100,320,0,

**Color**.*BLUE*, **Color**.*WHITE*, TileMode.*CLAMP*));

canvas.**drawRect**(220,0,320,100,Pnt);

// 가장자리 반복

Pnt.**setShader**(**new** **LinearGradient**(0,0,100,0,

**Color**.*BLUE*, **Color**.*WHITE*, TileMode.*CLAMP*));

canvas.**drawRect**(0,110,320,150,Pnt);

// 무늬 반복

Pnt.**setShader**(**new** **LinearGradient**(0,0,100,0,

**Color**.*BLUE*, **Color**.*WHITE*, TileMode.*REPEAT*));

canvas.**drawRect**(0,160,320,200,Pnt);

// 무늬 반사 반복

Pnt.**setShader**(**new** **LinearGradient**(0,0,100,0,

**Color**.*BLUE*, **Color**.*WHITE*, TileMode.*MIRROR*));

canvas.**drawRect**(0,210,320,250,Pnt);

// 여러 가지 색상 균등 배치

Pnt.**setShader**(**new** **LinearGradient**(0,0,320,0,

colors, **null**, TileMode.*CLAMP*));

canvas.**drawRect**(0,260,320,300,Pnt);

// 여러 가지 색상 임의 배치

Pnt.**setShader**(**new** **LinearGradient**(0,0,320,0,

colors, pos, TileMode.*CLAMP*));

canvas.**drawRect**(0,310,320,350,Pnt);

}

}

}

원형 그래디언트

**public** **class** **RadialGrad** **extends** **Activity** {

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

**setContentView**(**new** **MyView**(**this**));

}

**protected** **class** **MyView** **extends** **View** {

**public** **MyView**(**Context** context) {

**super**(context);

}

**public** **void** **onDraw**(**Canvas** canvas) {

**Paint** Pnt = **new** **Paint**();

Pnt.**setAntiAlias**(**true**);

**int**[] colors = { **Color**.*RED*, **Color**.*GREEN*, **Color**.*BLUE*, **Color**.*YELLOW*, **Color**.*WHITE* };

**float**[] pos = { 0.0f, 0.1f, 0.6f, 0.9f, 1.0f };

Pnt.**setShader**(**new** **RadialGradient**(80, 80, 70,

**Color**.*BLUE*, **Color**.*WHITE*, TileMode.*CLAMP*));

canvas.**drawCircle**(80, 80, 70, Pnt);

Pnt.**setShader**(**new** **RadialGradient**(230, 80, 70,

**Color**.*WHITE*, **Color**.*BLUE*, TileMode.*CLAMP*));

canvas.**drawCircle**(230, 80, 70, Pnt);

Pnt.**setShader**(**new** **RadialGradient**(80, 230, 70,

colors, **null**, TileMode.*CLAMP*));

canvas.**drawCircle**(80, 230, 70, Pnt);

Pnt.**setShader**(**new** **RadialGradient**(230, 230, 70,

colors, pos, TileMode.*CLAMP*));

canvas.**drawCircle**(230, 230, 70, Pnt);

}

}

}

비트맵 셰이더

**public** **class** **BitmapSdr** **extends** **Activity** {

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

**setContentView**(**new** **MyView**(**this**));

}

**protected** **class** **MyView** **extends** **View** {

**public** **MyView**(**Context** context) {

**super**(context);

}

**public** **void** **onDraw**(**Canvas** canvas) {

**Paint** Pnt = **new** **Paint**();

Pnt.**setAntiAlias**(**true**);

**Bitmap** clover = **BitmapFactory**.*decodeResource*(**getContext**().**getResources**(),

**R**.**drawable**.*clover*);

Pnt.**setShader**(**new** **BitmapShader**(clover, TileMode.*REPEAT*, TileMode.*REPEAT*));

canvas.**drawRect**(0, 0, 320, 150, Pnt);

Pnt.**setShader**(**new** **BitmapShader**(clover, TileMode.*MIRROR*, TileMode.*REPEAT*));

canvas.**drawRect**(0, 160, 320, 310, Pnt);

}

}

}

반복적인…

회전

**public** **class** **Rotate** **extends** **Activity** {

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

**setContentView**(**new** **MyView**(**this**));

}

**protected** **class** **MyView** **extends** **View** {

**public** **MyView**(**Context** context) {

**super**(context);

}

**public** **void** **onDraw**(**Canvas** canvas) {

**Paint** Pnt = **new** **Paint**();

Pnt.**setAntiAlias**(**true**);

Pnt.**setColor**(**Color**.*YELLOW*);

canvas.**drawRect**(100,100,200,200,Pnt);

Pnt.**setColor**(**Color**.*GREEN*);

canvas.**save**();

canvas.**rotate**(-30);

canvas.**drawRect**(100,100,200,200,Pnt);

Pnt.**setColor**(**Color**.*BLUE*);

canvas.**restore**();

canvas.**rotate**(30,150,150);

canvas.**drawRect**(100,100,200,200,Pnt);

}

}

}

평행이동.

canvas.**translate**(0

센서

**package** com.sensor;

**import** android.app.Activity;

**import** android.hardware.Sensor;

**import** android.hardware.SensorEvent;

**import** android.hardware.SensorEventListener;

**import** android.hardware.SensorManager;

**import** android.os.Bundle;

**import** android.util.Log;

**import** android.widget.TextView;

**public** **class** **CustomSensor** **extends** **Activity** {

**SensorManager** sm;

SensorEventListener accL;

SensorEventListener oriL;

**Sensor** oriSensor;

**Sensor** accSensor;

**TextView** ax, ay, az;

**TextView** ox, oy, oz;

@Override

**public** **void** **onCreate**(**Bundle** savedInstanceState) {

**super**.**onCreate**(savedInstanceState);

**setContentView**(**R**.**layout**.*main*);

sm = (**SensorManager**)**getSystemService**(*SENSOR\_SERVICE*); // SensorManager 인스턴스를 가져옴

oriSensor = sm.**getDefaultSensor**(**Sensor**.*TYPE\_ORIENTATION*); // 방향 센서

accSensor = sm.**getDefaultSensor**(**Sensor**.*TYPE\_ACCELEROMETER*); // 가속도 센서

oriL = **new** **oriListener**(); // 방향 센서 리스너 인스턴스

accL = **new** **accListener**(); // 가속도 센서 리스너 인스턴스

ax = (**TextView**)**findViewById**(**R**.**id**.*acc\_x*);

ay = (**TextView**)**findViewById**(**R**.**id**.*acc\_y*);

az = (**TextView**)**findViewById**(**R**.**id**.*acc\_z*);

ox = (**TextView**)**findViewById**(**R**.**id**.*ori\_x*);

oy = (**TextView**)**findViewById**(**R**.**id**.*ori\_y*);

oz = (**TextView**)**findViewById**(**R**.**id**.*ori\_z*);

}

@Override

**public** **void** **onResume**() {

**super**.**onResume**();

sm.**registerListener**(accL, accSensor, **SensorManager**.*SENSOR\_DELAY\_NORMAL*); // 가속도 센서 리스너 오브젝트를 등록

sm.**registerListener**(oriL, oriSensor, **SensorManager**.*SENSOR\_DELAY\_NORMAL*); // 방향 센서 리스너 오브젝트를 등록

}

@Override

**public** **void** **onPause**() {

**super**.**onPause**();

sm.**unregisterListener**(oriL); // unregister acceleration listener

sm.**unregisterListener**(accL); // unregister orientation listener

}

**private** **class** **accListener** **implements** SensorEventListener {

**public** **void** **onSensorChanged**(**SensorEvent** event) { // 가속도 센서 값이 바뀔때마다 호출됨

ax.**setText**(**Float**.*toString*(event.values[0]));

ay.**setText**(**Float**.*toString*(event.values[1]));

az.**setText**(**Float**.*toString*(event.values[2]));

**Log**.*i*("SENSOR", "Acceleration changed.");

**Log**.*i*("SENSOR", " Acceleration X: " + event.values[0]

+ ", Acceleration Y: " + event.values[1]

+ ", Acceleration Z: " + event.values[2]);

}

**public** **void** **onAccuracyChanged**(**Sensor** sensor, **int** accuracy) {

}

}

**private** **class** **oriListener** **implements** SensorEventListener {

**public** **void** **onSensorChanged**(**SensorEvent** event) { // 방향 센서 값이 바뀔때마다 호출됨

ox.**setText**(**Float**.*toString*(event.values[0]));

oy.**setText**(**Float**.*toString*(event.values[1]));

oz.**setText**(**Float**.*toString*(event.values[2]));

**Log**.*i*("SENSOR", "Orientation changed.");

**Log**.*i*("SENSOR", " Orientation X: " + event.values[0]

+ ", Orientation Y: " + event.values[1]

+ ", Orientation Z: " + event.values[2]);

}

**public** **void** **onAccuracyChanged**(**Sensor** sensor, **int** accuracy) {

}

}

}

# **List of ISO 639-1 codes**

From Wikipedia, the free encyclopedia

|  |  |
| --- | --- |
| http://upload.wikimedia.org/wikipedia/en/thumb/9/99/Question_book-new.svg/50px-Question_book-new.svg.png | This article **does not**[**cite**](http://en.wikipedia.org/wiki/Wikipedia:Citing_sources)**any**[**references or sources**](http://en.wikipedia.org/wiki/Wikipedia:Verifiability). Please help [improve this article](http://en.wikipedia.org/w/index.php?title=List_of_ISO_639-1_codes&action=edit) by adding citations to [reliable sources](http://en.wikipedia.org/wiki/Wikipedia:Identifying_reliable_sources). Unsourced material may be [challenged](http://en.wikipedia.org/wiki/Template:Citation_needed) and [removed](http://en.wikipedia.org/wiki/Wikipedia:Verifiability#Burden_of_evidence). *(April 2011)* |

|  |  |
| --- | --- |
| [Mergefrom.svg](http://en.wikipedia.org/wiki/File:Mergefrom.svg) | It has been suggested that [*ISO 639-1 language matrix*](http://en.wikipedia.org/wiki/ISO_639-1_language_matrix) be [merged](http://en.wikipedia.org/wiki/Wikipedia:Merging) into this article or section. ([Discuss](http://en.wikipedia.org/wiki/Talk:List_of_ISO_639-1_codes)) |

[ISO 639](http://en.wikipedia.org/wiki/ISO_639) is a standardized nomenclature used to classify all known languages. Each language is assigned a 2-letter (639-1) and 3-letter (639-2 and 639-3), lowercase abbreviation, amended in later versions of the nomenclature. The system is highly useful for linguists and ethnographers to categorize the languages spoken on a regional basis, and to compute analysis in the field of [lexicostatistics](http://en.wikipedia.org/wiki/Lexicostatistics). ISO 639 has five code lists.

## [[edit](http://en.wikipedia.org/w/index.php?title=List_of_ISO_639-1_codes&action=edit&section=1)]Partial ISO 639 table

This table lists all of:

* [ISO 639-1](http://en.wikipedia.org/wiki/ISO_639-1): two-letter codes, one per language or [macrolanguage](http://en.wikipedia.org/wiki/Macrolanguage)

And some of:

* [ISO 639-2](http://en.wikipedia.org/wiki/ISO_639-2)/T: three-letter codes, for the same languages as 639-1
* [ISO 639-2](http://en.wikipedia.org/wiki/ISO_639-2)/B: three-letter codes, mostly the same as 639-2/T, but with some codes derived from English names rather than native names of languages. In the following table, these differing codes are highlighted in [**boldface**](http://en.wikipedia.org/wiki/Boldface).
* [ISO 639-3](http://en.wikipedia.org/wiki/ISO_639-3): three-letter codes, the same as 639-2/T for languages, but with distinct codes for each dialect of macrolanguages

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | [**Language family**](http://en.wikipedia.org/wiki/Language_family)**[↓](http://en.wikipedia.org/wiki/List_of_ISO_639-1_codes)** | **Language name[↓](http://en.wikipedia.org/wiki/List_of_ISO_639-1_codes)** | **Native name[↓](http://en.wikipedia.org/wiki/List_of_ISO_639-1_codes)** | [**639-1**](http://en.wikipedia.org/wiki/ISO_639-1)**[↓](http://en.wikipedia.org/wiki/List_of_ISO_639-1_codes)** | [**639-2**](http://en.wikipedia.org/wiki/ISO_639-2)**/T[↓](http://en.wikipedia.org/wiki/List_of_ISO_639-1_codes)** | **639-2/B[↓](http://en.wikipedia.org/wiki/List_of_ISO_639-1_codes)** | [**639-3**](http://en.wikipedia.org/wiki/ISO_639-3)**[↓](http://en.wikipedia.org/wiki/List_of_ISO_639-1_codes)** | **Notes[↓](http://en.wikipedia.org/wiki/List_of_ISO_639-1_codes)** |
|  | [Northwest Caucasian](http://en.wikipedia.org/wiki/Northwest_Caucasian_languages) | [Abkhaz](http://en.wikipedia.org/wiki/Abkhaz_language) | аҧсуа | ab | abk | abk | abk |  |
|  | [Afro-Asiatic](http://en.wikipedia.org/wiki/Afro-Asiatic_languages) | [Afar](http://en.wikipedia.org/wiki/Afar_language) | Afaraf | aa | aar | aar | aar |  |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Afrikaans](http://en.wikipedia.org/wiki/Afrikaans_language) | Afrikaans | af | afr | afr | afr |  |
|  | [Niger-Congo](http://en.wikipedia.org/wiki/Niger-Congo_languages) | [Akan](http://en.wikipedia.org/wiki/Akan_language) | Akan | ak | aka | aka | aka + [2](http://en.wikipedia.org/wiki/ISO_639_macrolanguage#aka) | [macrolanguage](http://en.wikipedia.org/wiki/Macrolanguage), [Twi](http://en.wikipedia.org/wiki/Twi_language) is [tw/twi], [Fanti](http://en.wikipedia.org/wiki/Fanti_language) is [fat] |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Albanian](http://en.wikipedia.org/wiki/Albanian_language) | Shqip | sq | sqi | **alb** | sqi +[4](http://en.wikipedia.org/wiki/ISO_639_macrolanguage#sqi) | [macrolanguage](http://en.wikipedia.org/wiki/Macrolanguage) |
|  | [Afro-Asiatic](http://en.wikipedia.org/wiki/Afro-Asiatic_languages) | [Amharic](http://en.wikipedia.org/wiki/Amharic_language) | አማርኛ | am | amh | amh | amh |  |
|  | [Afro-Asiatic](http://en.wikipedia.org/wiki/Afro-Asiatic_languages) | [Arabic](http://en.wikipedia.org/wiki/Arabic_language) | العربية | ar | ara | ara | ara +[30](http://en.wikipedia.org/wiki/ISO_639_macrolanguage#ara) | [macrolanguage](http://en.wikipedia.org/wiki/Macrolanguage), [Standard Arabic](http://en.wikipedia.org/wiki/Standard_Arabic) is [arb] |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Aragonese](http://en.wikipedia.org/wiki/Aragonese_language) | Aragonés | an | arg | arg | arg |  |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Armenian](http://en.wikipedia.org/wiki/Armenian_language) | Հայերեն | hy | hye | **arm** | hye |  |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Assamese](http://en.wikipedia.org/wiki/Assamese_language) | অসমীয়া | as | asm | asm | asm |  |
|  | [Northeast Caucasian](http://en.wikipedia.org/wiki/Northeast_Caucasian_languages) | [Avaric](http://en.wikipedia.org/wiki/Avar_language) | авар мацӀ, магӀарул мацӀ | av | ava | ava | ava |  |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Avestan](http://en.wikipedia.org/wiki/Avestan_language) | avesta | ae | ave | ave | ave | [ancient](http://en.wikipedia.org/wiki/Ancient_languages) |
|  | [Aymaran](http://en.wikipedia.org/wiki/Aymaran_languages) | [Aymara](http://en.wikipedia.org/wiki/Aymara_language) | aymar aru | ay | aym | aym | aym + [2](http://en.wikipedia.org/wiki/ISO_639_macrolanguage#aym) | [macrolanguage](http://en.wikipedia.org/wiki/Macrolanguage) |
|  | [Turkic](http://en.wikipedia.org/wiki/Turkic_languages) | [Azerbaijani](http://en.wikipedia.org/wiki/Azerbaijani_language) | azərbaycan dili | az | aze | aze | aze + [2](http://en.wikipedia.org/wiki/ISO_639_macrolanguage#aze) | [macrolanguage](http://en.wikipedia.org/wiki/Macrolanguage) |
|  | [Niger-Congo](http://en.wikipedia.org/wiki/Niger-Congo_languages) | [Bambara](http://en.wikipedia.org/wiki/Bambara_language) | bamanankan | bm | bam | bam | bam |  |
|  | [Turkic](http://en.wikipedia.org/wiki/Turkic_languages) | [Bashkir](http://en.wikipedia.org/wiki/Bashkir_language) | башҡорт теле | ba | bak | bak | bak |  |
|  | [Language isolate](http://en.wikipedia.org/wiki/Language_isolate) | [Basque](http://en.wikipedia.org/wiki/Basque_language) | euskara, euskera | eu | eus | **baq** | eus |  |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Belarusian](http://en.wikipedia.org/wiki/Belarusian_language) | Беларуская | be | bel | bel | bel |  |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Bengali](http://en.wikipedia.org/wiki/Bengali_language) | বাংলা | bn | ben | ben | ben |  |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Bihari](http://en.wikipedia.org/wiki/Bihari_languages) | भोजपुरी | bh | bih | bih | — | Collective language code for [Bhojpuri](http://en.wikipedia.org/wiki/Bhojpuri_language),[Magahi](http://en.wikipedia.org/wiki/Magahi_language), and [Maithili](http://en.wikipedia.org/wiki/Maithili_language) |
|  | [Creole](http://en.wikipedia.org/wiki/Creole_languages) | [Bislama](http://en.wikipedia.org/wiki/Bislama_language) | Bislama | bi | bis | bis | bis |  |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Bosnian](http://en.wikipedia.org/wiki/Bosnian_language) | bosanski jezik | bs | bos | bos | bos |  |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Breton](http://en.wikipedia.org/wiki/Breton_language) | brezhoneg | br | bre | bre | bre |  |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Bulgarian](http://en.wikipedia.org/wiki/Bulgarian_language) | български език | bg | bul | bul | bul |  |
|  | [Sino-Tibetan](http://en.wikipedia.org/wiki/Sino-Tibetan_languages) | [Burmese](http://en.wikipedia.org/wiki/Burmese_language) | ဗမာစာ | my | mya | **bur** | mya |  |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Catalan](http://en.wikipedia.org/wiki/Catalan_language); [Valencian](http://en.wikipedia.org/wiki/Valencian) | Català | ca | cat | cat | cat |  |
|  | [Austronesian](http://en.wikipedia.org/wiki/Austronesian_languages) | [Chamorro](http://en.wikipedia.org/wiki/Chamorro_language) | Chamoru | ch | cha | cha | cha |  |
|  | [Northeast Caucasian](http://en.wikipedia.org/wiki/Northeast_Caucasian_languages) | [Chechen](http://en.wikipedia.org/wiki/Chechen_language) | нохчийн мотт | ce | che | che | che |  |
|  | [Niger-Congo](http://en.wikipedia.org/wiki/Niger-Congo_languages) | [Chichewa](http://en.wikipedia.org/wiki/Chichewa_language); Chewa; Nyanja | chiCheŵa, chinyanja | ny | nya | nya | nya |  |
|  | [Sino-Tibetan](http://en.wikipedia.org/wiki/Sino-Tibetan_languages) | [Chinese](http://en.wikipedia.org/wiki/Chinese_language) | 中文 (Zhōngwén), 汉语, 漢語 | zh | zho | **chi** | zho + [13](http://en.wikipedia.org/wiki/ISO_639_macrolanguage#zho) | [macrolanguage](http://en.wikipedia.org/wiki/Macrolanguage) |
|  | [Turkic](http://en.wikipedia.org/wiki/Turkic_languages) | [Chuvash](http://en.wikipedia.org/wiki/Chuvash_language) | чӑваш чӗлхи | cv | chv | chv | chv |  |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Cornish](http://en.wikipedia.org/wiki/Cornish_language) | Kernewek | kw | cor | cor | cor |  |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Corsican](http://en.wikipedia.org/wiki/Corsican_language) | corsu, lingua corsa | co | cos | cos | cos |  |
|  | [Algonquian](http://en.wikipedia.org/wiki/Algonquian_languages) | [Cree](http://en.wikipedia.org/wiki/Cree_language) | ᓀᐦᐃᔭᐍᐏᐣ | cr | cre | cre | cre +[6](http://en.wikipedia.org/wiki/ISO_639_macrolanguage#cre) | [macrolanguage](http://en.wikipedia.org/wiki/Macrolanguage) |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Croatian](http://en.wikipedia.org/wiki/Croatian_language) | hrvatski | hr | hrv | hrv | hrv |  |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Czech](http://en.wikipedia.org/wiki/Czech_language) | česky, čeština | cs | ces | **cze** | ces |  |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Danish](http://en.wikipedia.org/wiki/Danish_language) | dansk | da | dan | dan | dan |  |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Divehi](http://en.wikipedia.org/wiki/Dhivehi_language); Dhivehi; Maldivian; | ދިވެހި | dv | div | div | div |  |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Dutch](http://en.wikipedia.org/wiki/Dutch_language) | Nederlands, Vlaams | nl | nld | **dut** | nld |  |
|  | [Sino-Tibetan](http://en.wikipedia.org/wiki/Sino-Tibetan_languages) | [Dzongkha](http://en.wikipedia.org/wiki/Dzongkha_language) | རྫོང་ཁ | dz | dzo | dzo | dzo |  |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [English](http://en.wikipedia.org/wiki/English_language) | English | en | eng | eng | eng |  |
|  | [Constructed](http://en.wikipedia.org/wiki/Constructed_languages) | [Esperanto](http://en.wikipedia.org/wiki/Esperanto) | Esperanto | eo | epo | epo | epo | [constructed](http://en.wikipedia.org/wiki/Constructed_languages) |
|  | [Uralic](http://en.wikipedia.org/wiki/Uralic_languages) | [Estonian](http://en.wikipedia.org/wiki/Estonian_language) | eesti, eesti keel | et | est | est | est +[2](http://en.wikipedia.org/wiki/ISO_639_macrolanguage#est) | [macrolanguage](http://en.wikipedia.org/wiki/Macrolanguage) |
|  | [Niger-Congo](http://en.wikipedia.org/wiki/Niger-Congo_languages) | [Ewe](http://en.wikipedia.org/wiki/Ewe_language) | Eʋegbe | ee | ewe | ewe | ewe |  |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Faroese](http://en.wikipedia.org/wiki/Faroese_language) | føroyskt | fo | fao | fao | fao |  |
|  | [Austronesian](http://en.wikipedia.org/wiki/Austronesian_languages) | [Fijian](http://en.wikipedia.org/wiki/Fijian_language) | vosa Vakaviti | fj | fij | fij | fij |  |
|  | [Uralic](http://en.wikipedia.org/wiki/Uralic_languages) | [Finnish](http://en.wikipedia.org/wiki/Finnish_language) | suomi, suomen kieli | fi | fin | fin | fin |  |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [French](http://en.wikipedia.org/wiki/French_language) | français, langue française | fr | fra | **fre** | fra |  |
|  | [Niger-Congo](http://en.wikipedia.org/wiki/Niger-Congo_languages) | [Fula](http://en.wikipedia.org/wiki/Fula_language); Fulah; Pulaar; Pular | Fulfulde, Pulaar, Pular | ff | ful | ful | ful +[9](http://en.wikipedia.org/wiki/ISO_639_macrolanguage#ful) | [macrolanguage](http://en.wikipedia.org/wiki/Macrolanguage) |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Galician](http://en.wikipedia.org/wiki/Galician_language) | Galego | gl | glg | glg | glg |  |
|  | [South Caucasian](http://en.wikipedia.org/wiki/South_Caucasian_languages) | [Georgian](http://en.wikipedia.org/wiki/Georgian_language) | ქართული | ka | kat | **geo** | kat |  |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [German](http://en.wikipedia.org/wiki/German_language) | Deutsch | de | deu | **ger** | deu |  |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Greek](http://en.wikipedia.org/wiki/Greek_language), Modern | Ελληνικά | el | ell | **gre** | ell |  |
|  | [Tupi-Guarani](http://en.wikipedia.org/wiki/Tupi-Guarani_languages) | [Guaraní](http://en.wikipedia.org/wiki/Guaran%C3%AD_language) | Avañe'ẽ | gn | grn | grn | grn +[5](http://en.wikipedia.org/wiki/ISO_639_macrolanguage#grn) | [macrolanguage](http://en.wikipedia.org/wiki/Macrolanguage) |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Gujarati](http://en.wikipedia.org/wiki/Gujarati_language) | ગુજરાતી | gu | guj | guj | guj |  |
|  | [Creole](http://en.wikipedia.org/wiki/Creole_languages) | [Haitian](http://en.wikipedia.org/wiki/Haitian_Creole_language); Haitian Creole | Kreyòl ayisyen | ht | hat | hat | hat |  |
|  | [Afro-Asiatic](http://en.wikipedia.org/wiki/Afro-Asiatic_languages) | [Hausa](http://en.wikipedia.org/wiki/Hausa_language) | Hausa, هَوُسَ | ha | hau | hau | hau |  |
|  | [Afro-Asiatic](http://en.wikipedia.org/wiki/Afro-Asiatic_languages) | [Hebrew](http://en.wikipedia.org/wiki/Hebrew_language) (modern) | עברית | he | heb | heb | heb |  |
|  | [Niger-Congo](http://en.wikipedia.org/wiki/Niger-Congo_languages) | [Herero](http://en.wikipedia.org/wiki/Herero_language) | Otjiherero | hz | her | her | her |  |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Hindi](http://en.wikipedia.org/wiki/Hindi) | हिन्दी, हिंदी | hi | hin | hin | hin |  |
|  | [Austronesian](http://en.wikipedia.org/wiki/Austronesian_languages) | [Hiri Motu](http://en.wikipedia.org/wiki/Hiri_Motu_language) | Hiri Motu | ho | hmo | hmo | hmo |  |
|  | [Uralic](http://en.wikipedia.org/wiki/Uralic_languages) | [Hungarian](http://en.wikipedia.org/wiki/Hungarian_language) | Magyar | hu | hun | hun | hun |  |
|  | [Constructed](http://en.wikipedia.org/wiki/Constructed_languages) | [Interlingua](http://en.wikipedia.org/wiki/Interlingua) | Interlingua | ia | ina | ina | ina | constructed by [International Auxiliary Language Association](http://en.wikipedia.org/wiki/International_Auxiliary_Language_Association) |
|  | [Austronesian](http://en.wikipedia.org/wiki/Austronesian_languages) | [Indonesian](http://en.wikipedia.org/wiki/Indonesian_language) | Bahasa Indonesia | id | ind | ind | ind | Covered by macrolanguage [ms/msa] |
|  | [Constructed](http://en.wikipedia.org/wiki/Constructed_languages) | [Interlingue](http://en.wikipedia.org/wiki/Interlingue_language) | Originally called [Occidental](http://en.wikipedia.org/wiki/Occidental_language); then Interlingue after WWII | ie | ile | ile | ile | constructed by [Edgar de Wahl](http://en.wikipedia.org/wiki/Edgar_de_Wahl), first published in 1922 |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Irish](http://en.wikipedia.org/wiki/Irish_language) | Gaeilge | ga | gle | gle | gle |  |
|  | [Niger-Congo](http://en.wikipedia.org/wiki/Niger-Congo_languages) | [Igbo](http://en.wikipedia.org/wiki/Igbo_language) | Asụsụ Igbo | ig | ibo | ibo | ibo |  |
|  | [Eskimo-Aleut](http://en.wikipedia.org/wiki/Eskimo-Aleut_languages) | [Inupiaq](http://en.wikipedia.org/wiki/Inupiaq_language) | Iñupiaq, Iñupiatun | ik | ipk | ipk | ipk +[2](http://en.wikipedia.org/wiki/ISO_639_macrolanguage#ipk) | [macrolanguage](http://en.wikipedia.org/wiki/Macrolanguage) |
|  | [Constructed](http://en.wikipedia.org/wiki/Constructed_languages) | [Ido](http://en.wikipedia.org/wiki/Ido) | Ido | io | ido | ido | ido |  |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Icelandic](http://en.wikipedia.org/wiki/Icelandic_language) | Íslenska | is | isl | **ice** | isl |  |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Italian](http://en.wikipedia.org/wiki/Italian_language) | Italiano | it | ita | ita | ita |  |
|  | [Eskimo-Aleut](http://en.wikipedia.org/wiki/Eskimo-Aleut_languages) | [Inuktitut](http://en.wikipedia.org/wiki/Inuktitut) | ᐃᓄᒃᑎᑐᑦ | iu | iku | iku | iku +[2](http://en.wikipedia.org/wiki/ISO_639_macrolanguage#iku) | macrolanguage |
|  | [Japonic](http://en.wikipedia.org/wiki/Japonic) | [Japanese](http://en.wikipedia.org/wiki/Japanese_language) | 日本語 (にほんご) | ja | jpn | jpn | jpn |  |
|  | [Austronesian](http://en.wikipedia.org/wiki/Austronesian_languages) | [Javanese](http://en.wikipedia.org/wiki/Javanese_language) | basa Jawa | jv | jav | jav | jav |  |
|  | [Eskimo-Aleut](http://en.wikipedia.org/wiki/Eskimo-Aleut_languages) | [Kalaallisut](http://en.wikipedia.org/wiki/Kalaallisut_language), Greenlandic | kalaallisut, kalaallit oqaasii | kl | kal | kal | kal |  |
|  | [Dravidian](http://en.wikipedia.org/wiki/Dravidian_languages) | [Kannada](http://en.wikipedia.org/wiki/Kannada_language) | ಕನ್ನಡ | kn | kan | kan | kan |  |
|  | [Nilo-Saharan](http://en.wikipedia.org/wiki/Nilo-Saharan_languages) | [Kanuri](http://en.wikipedia.org/wiki/Kanuri_language) | Kanuri | kr | kau | kau | kau + [3](http://en.wikipedia.org/wiki/ISO_639_macrolanguage#kau) | [macrolanguage](http://en.wikipedia.org/wiki/Macrolanguage) |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Kashmiri](http://en.wikipedia.org/wiki/Kashmiri_language) | कश्मीरी, كشميري‎ | ks | kas | kas | kas |  |
|  | [Turkic](http://en.wikipedia.org/wiki/Turkic_languages) | [Kazakh](http://en.wikipedia.org/wiki/Kazakh_language) | Қазақ тілі | kk | kaz | kaz | kaz |  |
|  | [Austro-Asiatic](http://en.wikipedia.org/wiki/Austro-Asiatic_languages) | [Khmer](http://en.wikipedia.org/wiki/Khmer_language) | ភាសាខ្មែរ | km | khm | khm | khm | a.k.a. Cambodian |
|  | [Niger-Congo](http://en.wikipedia.org/wiki/Niger-Congo_languages) | [Kikuyu](http://en.wikipedia.org/wiki/Gikuyu_language), Gikuyu | Gĩkũyũ | ki | kik | kik | kik |  |
|  | [Niger-Congo](http://en.wikipedia.org/wiki/Niger-Congo_languages) | [Kinyarwanda](http://en.wikipedia.org/wiki/Kinyarwanda) | Ikinyarwanda | rw | kin | kin | kin |  |
|  | [Turkic](http://en.wikipedia.org/wiki/Turkic_languages) | [Kirghiz](http://en.wikipedia.org/wiki/Kyrgyz_language), Kyrgyz | кыргыз тили | ky | kir | kir | kir |  |
|  | [Uralic](http://en.wikipedia.org/wiki/Uralic_languages) | [Komi](http://en.wikipedia.org/wiki/Komi_language) | коми кыв | kv | kom | kom | kom + [2](http://en.wikipedia.org/wiki/ISO_639_macrolanguage#kom) | macrolanguage |
|  | [Niger-Congo](http://en.wikipedia.org/wiki/Niger-Congo_languages) | [Kongo](http://en.wikipedia.org/wiki/Kongo_language) | KiKongo | kg | kon | kon | kon + [3](http://en.wikipedia.org/wiki/ISO_639_macrolanguage#kon) | macrolanguage |
|  | [Language isolate](http://en.wikipedia.org/wiki/Language_isolate) | [Korean](http://en.wikipedia.org/wiki/Korean_language) | 한국어 (韓國語), 조선어 (朝鮮語) | ko | kor | kor | kor |  |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Kurdish](http://en.wikipedia.org/wiki/Kurdish_language) | Kurdî, كوردی‎ | ku | kur | kur | kur +[3](http://en.wikipedia.org/wiki/ISO_639_macrolanguage#kur) | macrolanguage |
|  | [Niger-Congo](http://en.wikipedia.org/wiki/Niger-Congo_languages) | [Kwanyama](http://en.wikipedia.org/wiki/Kwanyama), Kuanyama | Kuanyama | kj | kua | kua | kua |  |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Latin](http://en.wikipedia.org/wiki/Latin) | latine, lingua latina | la | lat | lat | lat | ancient |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Luxembourgish](http://en.wikipedia.org/wiki/Luxembourgish_language), Letzeburgesch | Lëtzebuergesch | lb | ltz | ltz | ltz |  |
|  | [Niger-Congo](http://en.wikipedia.org/wiki/Niger-Congo_languages) | [Luganda](http://en.wikipedia.org/wiki/Luganda) | Luganda | lg | lug | lug | lug |  |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Limburgish](http://en.wikipedia.org/wiki/Limburgish_language), Limburgan, Limburger | Limburgs | li | lim | lim | lim |  |
|  | [Niger-Congo](http://en.wikipedia.org/wiki/Niger-Congo_languages) | [Lingala](http://en.wikipedia.org/wiki/Lingala_language) | Lingála | ln | lin | lin | lin |  |
|  | [Tai-Kadai](http://en.wikipedia.org/wiki/Tai-Kadai_languages) | [Lao](http://en.wikipedia.org/wiki/Lao_language) | ພາສາລາວ | lo | lao | lao | lao |  |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Lithuanian](http://en.wikipedia.org/wiki/Lithuanian_language) | lietuvių kalba | lt | lit | lit | lit |  |
|  | [Niger-Congo](http://en.wikipedia.org/wiki/Niger-Congo_languages) | [Luba-Katanga](http://en.wikipedia.org/wiki/Tshiluba_language) |  | lu | lub | lub | lub |  |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Latvian](http://en.wikipedia.org/wiki/Latvian_language) | latviešu valoda | lv | lav | lav | lav |  |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Manx](http://en.wikipedia.org/wiki/Manx_language) | Gaelg, Gailck | gv | glv | glv | glv |  |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Macedonian](http://en.wikipedia.org/wiki/Macedonian_language) | македонски јазик | mk | mkd | **mac** | mkd |  |
|  | [Austronesian](http://en.wikipedia.org/wiki/Austronesian_languages) | [Malagasy](http://en.wikipedia.org/wiki/Malagasy_language) | Malagasy fiteny | mg | mlg | mlg | mlg + [10](http://en.wikipedia.org/wiki/ISO_639_macrolanguage#mlg) | macrolanguage |
|  | [Austronesian](http://en.wikipedia.org/wiki/Austronesian_languages) | [Malay](http://en.wikipedia.org/wiki/Malay_language) | bahasa Melayu, بهاس ملايو‎ | ms | msa | **may** | msa + [13](http://en.wikipedia.org/wiki/ISO_639_macrolanguage#msa) | macrolanguage, [Standard Malay](http://en.wikipedia.org/wiki/Malaysian_language) is [zsm],[Indonesian](http://en.wikipedia.org/wiki/Indonesian_language) is [id/ind] |
|  | [Dravidian](http://en.wikipedia.org/wiki/Dravidian_languages) | [Malayalam](http://en.wikipedia.org/wiki/Malayalam_language) | മലയാളം | ml | mal | mal | mal |  |
|  | [Afro-Asiatic](http://en.wikipedia.org/wiki/Afro-Asiatic_languages) | [Maltese](http://en.wikipedia.org/wiki/Maltese_language) | Malti | mt | mlt | mlt | mlt |  |
|  | [Austronesian](http://en.wikipedia.org/wiki/Austronesian_languages) | [Māori](http://en.wikipedia.org/wiki/M%C4%81ori_language) | te reo Māori | mi | mri | **mao** | mri |  |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Marathi (Marāṭhī)](http://en.wikipedia.org/wiki/Marathi_language) | मराठी | mr | mar | mar | mar |  |
|  | [Austronesian](http://en.wikipedia.org/wiki/Austronesian_languages) | [Marshallese](http://en.wikipedia.org/wiki/Marshallese_language) | Kajin M̧ajeļ | mh | mah | mah | mah |  |
|  | [Mongolic](http://en.wikipedia.org/wiki/Mongolic_languages) | [Mongolian](http://en.wikipedia.org/wiki/Mongolian_language) | монгол | mn | mon | mon | mon + [2](http://en.wikipedia.org/wiki/ISO_639_macrolanguage#mon) | macrolanguage |
|  | [Austronesian](http://en.wikipedia.org/wiki/Austronesian_languages) | [Nauru](http://en.wikipedia.org/wiki/Nauruan_language) | Ekakairũ Naoero | na | nau | nau | nau |  |
|  | [Dene-Yeniseian](http://en.wikipedia.org/wiki/Dene-Yeniseian_languages) | [Navajo](http://en.wikipedia.org/wiki/Navajo_language), Navaho | Diné bizaad, Dinékʼehǰí | nv | nav | nav | nav |  |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Norwegian Bokmål](http://en.wikipedia.org/wiki/Bokm%C3%A5l) | Norsk bokmål | nb | nob | nob | nob | Covered by macrolanguage [no/nor] |
|  | [Niger-Congo](http://en.wikipedia.org/wiki/Niger-Congo_languages) | [North Ndebele](http://en.wikipedia.org/wiki/Northern_Ndebele_language) | isiNdebele | nd | nde | nde | nde |  |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Nepali](http://en.wikipedia.org/wiki/Nepali_language) | नेपाली | ne | nep | nep | nep |  |
|  | [Niger-Congo](http://en.wikipedia.org/wiki/Niger-Congo_languages) | [Ndonga](http://en.wikipedia.org/wiki/Ndonga) | Owambo | ng | ndo | ndo | ndo |  |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Norwegian Nynorsk](http://en.wikipedia.org/wiki/Nynorsk) | Norsk nynorsk | nn | nno | nno | nno | Covered by macrolanguage [no/nor] |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Norwegian](http://en.wikipedia.org/wiki/Norwegian_language) | Norsk | no | nor | nor | nor +[2](http://en.wikipedia.org/wiki/ISO_639_macrolanguage#nor) | macrolanguage, [Bokmål](http://en.wikipedia.org/wiki/Bokm%C3%A5l) is [nb/nob], [Nynorsk](http://en.wikipedia.org/wiki/Nynorsk)is [nn/nno] |
|  | [Sino-Tibetan](http://en.wikipedia.org/wiki/Sino-Tibetan_languages) | [Nuosu](http://en.wikipedia.org/wiki/Nuosu_language) | ꆈꌠ꒿ Nuosuhxop | ii | iii | iii | iii | Standard form of [Yi languages](http://en.wikipedia.org/wiki/Yi_language) |
|  | [Niger-Congo](http://en.wikipedia.org/wiki/Niger-Congo_languages) | [South Ndebele](http://en.wikipedia.org/wiki/Southern_Ndebele_language) | isiNdebele | nr | nbl | nbl | nbl |  |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Occitan](http://en.wikipedia.org/wiki/Occitan_language) | Occitan | oc | oci | oci | oci |  |
|  | [Algonquian](http://en.wikipedia.org/wiki/Algonquian_languages) | [Ojibwe](http://en.wikipedia.org/wiki/Ojibwe_language), Ojibwa | ᐊᓂᔑᓈᐯᒧᐎᓐ | oj | oji | oji | oji +[7](http://en.wikipedia.org/wiki/ISO_639_macrolanguage#oji) | macrolanguage |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Old Church Slavonic](http://en.wikipedia.org/wiki/Old_Church_Slavonic), Church Slavic, Church Slavonic, Old Bulgarian, Old Slavonic | ѩзыкъ словѣньскъ | cu | chu | chu | chu | [ancient](http://en.wikipedia.org/wiki/Ancient_languages) |
|  | [Afro-Asiatic](http://en.wikipedia.org/wiki/Afro-Asiatic_languages) | [Oromo](http://en.wikipedia.org/wiki/Oromo_language) | Afaan Oromoo | om | orm | orm | orm + [4](http://en.wikipedia.org/wiki/ISO_639_macrolanguage#orm) | macrolanguage |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Oriya](http://en.wikipedia.org/wiki/Oriya_language) | ଓଡ଼ିଆ | or | ori | ori | ori |  |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Ossetian](http://en.wikipedia.org/wiki/Ossetic_language), Ossetic | ирон æвзаг | os | oss | oss | oss |  |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Panjabi](http://en.wikipedia.org/wiki/Punjabi_language), Punjabi | ਪੰਜਾਬੀ, پنجابی‎ | pa | pan | pan | pan |  |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Pāli](http://en.wikipedia.org/wiki/P%C4%81li_language) | पाऴि | pi | pli | pli | pli | ancient |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Persian](http://en.wikipedia.org/wiki/Persian_language) | فارسی | fa | fas | **per** | fas +[2](http://en.wikipedia.org/wiki/ISO_639_macrolanguage#fas) | macrolanguage |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Polish](http://en.wikipedia.org/wiki/Polish_language) | polski | pl | pol | pol | pol |  |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Pashto](http://en.wikipedia.org/wiki/Pashto_language), Pushto | پښتو | ps | pus | pus | pus + [3](http://en.wikipedia.org/wiki/ISO_639_macrolanguage#pus) | macrolanguage |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Portuguese](http://en.wikipedia.org/wiki/Portuguese_language) | Português | pt | por | por | por |  |
|  | [Quechuan](http://en.wikipedia.org/wiki/Quechuan_languages) | [Quechua](http://en.wikipedia.org/wiki/Quechua_language) | Runa Simi, Kichwa | qu | que | que | que + [44](http://en.wikipedia.org/wiki/ISO_639_macrolanguage#que) | macrolanguage |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Romansh](http://en.wikipedia.org/wiki/Romansh_language) | rumantsch grischun | rm | roh | roh | roh |  |
|  | [Niger-Congo](http://en.wikipedia.org/wiki/Niger-Congo_languages) | [Kirundi](http://en.wikipedia.org/wiki/Kirundi) | Ikirundi | rn | run | run | run |  |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Romanian](http://en.wikipedia.org/wiki/Romanian_language), [Moldavian](http://en.wikipedia.org/wiki/Moldavian_language), Moldovan | română | ro | ron | **rum** | ron | [mo] for Moldavian has been withdrawn, recommending [ro] also for Moldavian |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Russian](http://en.wikipedia.org/wiki/Russian_language) | русский язык | ru | rus | rus | rus |  |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Sanskrit (Saṁskṛta)](http://en.wikipedia.org/wiki/Sanskrit) | संस्कृतम् | sa | san | san | san | ancient, still spoken |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Sardinian](http://en.wikipedia.org/wiki/Sardinian_language) | sardu | sc | srd | srd | srd +[4](http://en.wikipedia.org/wiki/ISO_639_macrolanguage#srd) | macrolanguage |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Sindhi](http://en.wikipedia.org/wiki/Sindhi_language) | सिन्धी, سنڌي، سندھی‎ | sd | snd | snd | snd |  |
|  | [Uralic](http://en.wikipedia.org/wiki/Uralic_languages) | [Northern Sami](http://en.wikipedia.org/wiki/Northern_Sami) | Davvisámegiella | se | sme | sme | sme |  |
|  | [Austronesian](http://en.wikipedia.org/wiki/Austronesian_Languages) | [Samoan](http://en.wikipedia.org/wiki/Samoan_language) | gagana fa'a Samoa | sm | smo | smo | smo |  |
|  | [Creole](http://en.wikipedia.org/wiki/Creole_languages) | [Sango](http://en.wikipedia.org/wiki/Sango_language) | yângâ tî sängö | sg | sag | sag | sag |  |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Serbian](http://en.wikipedia.org/wiki/Serbian_language) | српски језик | sr | srp | srp | srp |  |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Scottish Gaelic](http://en.wikipedia.org/wiki/Scottish_Gaelic_language); Gaelic | Gàidhlig | gd | gla | gla | gla |  |
|  | [Niger-Congo](http://en.wikipedia.org/wiki/Niger-Congo_languages) | [Shona](http://en.wikipedia.org/wiki/Shona_language) | chiShona | sn | sna | sna | sna |  |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Sinhala](http://en.wikipedia.org/wiki/Sinhala_language), Sinhalese | සිංහල | si | sin | sin | sin |  |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Slovak](http://en.wikipedia.org/wiki/Slovak_language) | slovenčina | sk | slk | **slo** | slk |  |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Slovene](http://en.wikipedia.org/wiki/Slovene_language) | slovenščina | sl | slv | slv | slv |  |
|  | [Afro-Asiatic](http://en.wikipedia.org/wiki/Afro-Asiatic_languages) | [Somali](http://en.wikipedia.org/wiki/Somali_language) | Soomaaliga, af Soomaali | so | som | som | som |  |
|  | [Niger-Congo](http://en.wikipedia.org/wiki/Niger-Congo_languages) | [Southern Sotho](http://en.wikipedia.org/wiki/Sotho_language) | Sesotho | st | sot | sot | sot |  |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Spanish](http://en.wikipedia.org/wiki/Spanish_language); Castilian | español, castellano | es | spa | spa | spa |  |
|  | [Austronesian](http://en.wikipedia.org/wiki/Austronesian_languages) | [Sundanese](http://en.wikipedia.org/wiki/Sundanese_language) | Basa Sunda | su | sun | sun | sun |  |
|  | [Niger-Congo](http://en.wikipedia.org/wiki/Niger-Congo_languages) | [Swahili](http://en.wikipedia.org/wiki/Swahili_language) | Kiswahili | sw | swa | swa | swa + [2](http://en.wikipedia.org/wiki/ISO_639_macrolanguage#swa) | macrolanguage |
|  | [Niger-Congo](http://en.wikipedia.org/wiki/Niger-Congo_languages) | [Swati](http://en.wikipedia.org/wiki/Swati_language) | SiSwati | ss | ssw | ssw | ssw |  |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Swedish](http://en.wikipedia.org/wiki/Swedish_language) | svenska | sv | swe | swe | swe |  |
|  | [Dravidian](http://en.wikipedia.org/wiki/Dravidian_languages) | [Tamil](http://en.wikipedia.org/wiki/Tamil_language) | தமிழ் | ta | tam | tam | tam |  |
|  | [Dravidian](http://en.wikipedia.org/wiki/Dravidian_languages) | [Telugu](http://en.wikipedia.org/wiki/Telugu_language) | తెలుగు | te | tel | tel | tel |  |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Tajik](http://en.wikipedia.org/wiki/Tajik_language) | тоҷикӣ, toğikī, تاجیکی‎ | tg | tgk | tgk | tgk |  |
|  | [Tai-Kadai](http://en.wikipedia.org/wiki/Tai-Kadai_languages) | [Thai](http://en.wikipedia.org/wiki/Thai_language) | ไทย | th | tha | tha | tha |  |
|  | [Afro-Asiatic](http://en.wikipedia.org/wiki/Afro-Asiatic_languages) | [Tigrinya](http://en.wikipedia.org/wiki/Tigrinya_language) | ትግርኛ | ti | tir | tir | tir |  |
|  | [Sino-Tibetan](http://en.wikipedia.org/wiki/Sino-Tibetan_languages) | [Tibetan Standard](http://en.wikipedia.org/wiki/Standard_Tibetan), Tibetan, Central | བོད་ཡིག | bo | bod | **tib** | bod |  |
|  | [Turkic](http://en.wikipedia.org/wiki/Turkic_languages) | [Turkmen](http://en.wikipedia.org/wiki/Turkmen_language) | Türkmen, Түркмен | tk | tuk | tuk | tuk |  |
|  | [Austronesian](http://en.wikipedia.org/wiki/Austronesian_languages) | [Tagalog](http://en.wikipedia.org/wiki/Tagalog_language) | Wikang Tagalog, ᜏᜒᜃᜅ᜔ ᜆᜄᜎᜓᜄ᜔ | tl | tgl | tgl | tgl | Note: [Filipino](http://en.wikipedia.org/wiki/Filipino_language) (Pilipino) has the code [fil] |
|  | [Niger-Congo](http://en.wikipedia.org/wiki/Niger-Congo_languages) | [Tswana](http://en.wikipedia.org/wiki/Tswana_language) | Setswana | tn | tsn | tsn | tsn |  |
|  | [Austronesian](http://en.wikipedia.org/wiki/Austronesian_languages) | [Tonga](http://en.wikipedia.org/wiki/Tongan_language) (Tonga Islands) | faka Tonga | to | ton | ton | ton |  |
|  | [Turkic](http://en.wikipedia.org/wiki/Turkic_languages) | [Turkish](http://en.wikipedia.org/wiki/Turkish_language) | Türkçe | tr | tur | tur | tur |  |
|  | [Niger-Congo](http://en.wikipedia.org/wiki/Niger-Congo_languages) | [Tsonga](http://en.wikipedia.org/wiki/Tsonga_language) | Xitsonga | ts | tso | tso | tso |  |
|  | [Turkic](http://en.wikipedia.org/wiki/Turkic_languages) | [Tatar](http://en.wikipedia.org/wiki/Tatar_language) | татарча, tatarça, تاتارچا‎ | tt | tat | tat | tat |  |
|  | [Niger-Congo](http://en.wikipedia.org/wiki/Niger-Congo_languages) | [Twi](http://en.wikipedia.org/wiki/Twi) | Twi | tw | twi | twi | twi | Covered by macrolanguage [ak/aka] |
|  | [Austronesian](http://en.wikipedia.org/wiki/Austronesian_languages) | [Tahitian](http://en.wikipedia.org/wiki/Tahitian_language) | Reo Tahiti | ty | tah | tah | tah | One of the Reo Mā`ohi (languages of French Polynesia) |
|  | [Turkic](http://en.wikipedia.org/wiki/Turkic_languages) | [Uighur](http://en.wikipedia.org/wiki/Uyghur_language), Uyghur | Uyƣurqə, ئۇيغۇرچە‎ | ug | uig | uig | uig |  |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Ukrainian](http://en.wikipedia.org/wiki/Ukrainian_language) | українська | uk | ukr | ukr | ukr |  |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Urdu](http://en.wikipedia.org/wiki/Urdu) | اردو | ur | urd | urd | urd |  |
|  | [Turkic](http://en.wikipedia.org/wiki/Turkic_languages) | [Uzbek](http://en.wikipedia.org/wiki/Uzbek_language) | O'zbek, Ўзбек, أۇزبېك‎ | uz | uzb | uzb | uzb + [2](http://en.wikipedia.org/wiki/ISO_639_macrolanguage#uzb) | macrolanguage |
|  | [Niger-Congo](http://en.wikipedia.org/wiki/Niger-Congo_languages) | [Venda](http://en.wikipedia.org/wiki/Venda_language) | Tshivenḓa | ve | ven | ven | ven |  |
|  | [Austro-Asiatic](http://en.wikipedia.org/wiki/Austro-Asiatic_languages) | [Vietnamese](http://en.wikipedia.org/wiki/Vietnamese_language) | Tiếng Việt | vi | vie | vie | vie |  |
|  | [Constructed](http://en.wikipedia.org/wiki/Constructed_languages) | [Volapük](http://en.wikipedia.org/wiki/Volap%C3%BCk) | Volapük | vo | vol | vol | vol | constructed |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Walloon](http://en.wikipedia.org/wiki/Walloon_language) | Walon | wa | wln | wln | wln |  |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Welsh](http://en.wikipedia.org/wiki/Welsh_language) | Cymraeg | cy | cym | **wel** | cym |  |
|  | [Niger-Congo](http://en.wikipedia.org/wiki/Niger-Congo_languages) | [Wolof](http://en.wikipedia.org/wiki/Wolof_language) | Wollof | wo | wol | wol | wol |  |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Western Frisian](http://en.wikipedia.org/wiki/West_Frisian_language) | Frysk | fy | fry | fry | fry |  |
|  | [Niger-Congo](http://en.wikipedia.org/wiki/Niger-Congo_languages) | [Xhosa](http://en.wikipedia.org/wiki/Xhosa_language) | isiXhosa | xh | xho | xho | xho |  |
|  | [Indo-European](http://en.wikipedia.org/wiki/Indo-European_languages) | [Yiddish](http://en.wikipedia.org/wiki/Yiddish_language) | ייִדיש | yi | yid | yid | yid +[2](http://en.wikipedia.org/wiki/ISO_639_macrolanguage#yid) | macrolanguage |
|  | [Niger-Congo](http://en.wikipedia.org/wiki/Niger-Congo_languages) | [Yoruba](http://en.wikipedia.org/wiki/Yoruba_language) | Yorùbá | yo | yor | yor | yor |  |
|  | [Tai-Kadai](http://en.wikipedia.org/wiki/Tai-Kadai_languages) | [Zhuang](http://en.wikipedia.org/wiki/Zhuang_language), Chuang | Saɯ cueŋƅ, Saw cuengh | za | zha | zha | zha + [16](http://en.wikipedia.org/wiki/ISO_639_macrolanguage#zha) | macrolanguage |
|  | [Niger-Congo](http://en.wikipedia.org/wiki/Niger-Congo_languages) | [Zulu](http://en.wikipedia.org/wiki/Zulu_language) | isiZulu | zu | zul | zul | zul |  |

## [[edit](http://en.wikipedia.org/w/index.php?title=List_of_ISO_639-1_codes&action=edit&section=2)]See also

* [Unicode](http://en.wikipedia.org/wiki/Unicode)

## [[edit](http://en.wikipedia.org/w/index.php?title=List_of_ISO_639-1_codes&action=edit&section=3)]External links

* [Official web site for ISO 639](http://www.loc.gov/standards/iso639-2/)
* [Official ISO 639 list](http://www.loc.gov/standards/iso639-2/php/English_list.php)
* [meta:Template:List of language names ordered by code](http://meta.wikimedia.org/wiki/Template:List_of_language_names_ordered_by_code)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | [[hide](http://en.wikipedia.org/wiki/List_of_ISO_639-1_codes)][v](http://en.wikipedia.org/wiki/Template:Footer_ISO_639) **·** [d](http://en.wikipedia.org/w/index.php?title=Template_talk:Footer_ISO_639&action=edit&redlink=1) **·** [e](http://en.wikipedia.org/w/index.php?title=Template:Footer_ISO_639&action=edit)[**ISO 639**](http://en.wikipedia.org/wiki/ISO_639)**and**[**ISO 639 macrolanguage**](http://en.wikipedia.org/wiki/ISO_639_macrolanguage) | | |  |  | | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | |  | [ISO 639-1](http://en.wikipedia.org/wiki/ISO_639-1) **list of codes** languages | [ISO 639-2](http://en.wikipedia.org/wiki/ISO_639-2) [list of codes](http://en.wikipedia.org/wiki/List_of_ISO_639-2_codes) languages | [ISO 639-3](http://en.wikipedia.org/wiki/ISO_639-3) [list of codes](http://en.wikipedia.org/wiki/List_of_ISO_639-3_codes) languages | [ISO 639-4](http://en.wikipedia.org/wiki/ISO_639) — guidelines | [ISO 639-5](http://en.wikipedia.org/wiki/ISO_639-5) [list of codes](http://en.wikipedia.org/wiki/List_of_ISO_639-5_codes) families/groups | [ISO 639-6](http://en.wikipedia.org/wiki/ISO_639-6) — variants | | | |

다국어