DART - SOMO LA 1: KAZI ZA DART PROGRAMMING NA HISTORIA YAKE

Summary: Katika somo hili utakwenda kujifunza kuhusu kazi za dart language. Pia nimekuandalia istoria fupi ya lugha ya DART.

DART

Dart ni miongoni mwa general purpose programming language. Hizi ni lugha za kikompyuta ambazo hufanyabkazi maneno mengi kwa mfano kwenye web app, software, na maneno mengineyo.

Hii ni course ya dart yenye lengo la kukupa ufahamu Ili iwe rahisi kuelewa matumizi ya flutter. Katika course hii tutakwenda kujifunza lugha hii pamoja na kufanyia mazoezi.

Maandalizi ya somo:

1. **Kwa wanaotumia kompyuta**

1. Download software ya visual studio ya Microsoft ama webstorm ya jetbrainin
2. Nenda kwenye sehemu ya kionstall extensions tafuta dart Kisha download. Hakikisha una bando lisilopunguwa MB 500
3. Kisha download dart sdk hapa kisha fuata maelekezo
4. Baada ya hapo utaanza project mpya ya dart.

2. **Kwa watumiaji wa simu**

Download App ya dart compiler hapa

Baada ya hapo utaifunguwa na kuanza kuandika code.

Online compiler

Kama ukiona ni changamoto ku install dart sdk unaweza kufanya majaribio online kwenye dart compiler. Cheki hapa

Matumizi ya dart

Dart inaweza kutumia kwa ajili ya:-

1. Kutengeneza website
2. Kutengeneza web App

3. Kutengeneza mobile app
4. Hutumika kwenye server
5. Kutengeneza desktop software

Kwa kuwa ni general purpose inaweza kufanya mambo mengi zaidi.

Historia fupi Dart

Dart language ikianza kudizainiwa na Lars Bak na luendelezwa na kampuni ya Google. Kwa mara ya kwanza ilitambulishwa mwaka 2011.

Mwaka 2013 toleo la kwanza la dart lilitambulishwa kama **Dart 1.0** wakati huu kulitokea malalamiko mengi kuhusu lugha hii. Hivyo mwaka 2015 toleo la **Dart 1.9** likiwa limejikita sana katika ku compile dart kuwa JavaScript. Mwaka 2018 mabadiliko zaidi yakaongezwa hasa kwenye **type system** na toleo hili lilitwa **Dart 2.0** yaani toleo la pili.

Baada ya mabadiliko zaidi toleo la **2.6** lilikuja na extension ya **dart2native** ambayo iliwezesha ku compile dart kwenye linux, macOS na window. Katika kipindi hiki mahusiano ya flutter na dart yakazidi kuimarika.

Mpaka kufikia leo 24/10/2023 tupo toleo la 3.1.4 ambalo lilitolewa mwaka 2023 tarehe 18 mwenye wa 10.

Mwisho:

Tukutane somo linalofuata tutajifunza kuhusu sytax za dart.

...**Mkala hii imeandaliwa Bongoclass. Kupata makala zaidi tembelea website yetu www.bongoclass.com**...

DART SOMO LA 2: SYNTAX ZA DART

Summary: Katika somo hili tutatkwenda kuziona baadhi ya xyntax yaani kanunu za uandishi wa code za dart.

Somo letu litakwenda kuanzia kwenye program ya kwanza kabisa ya **Hello World** copi code hapo chini kisa zi pest kwenye dart editor inayotumia kisha bofya batani ya ku excute code ama ku run code.

```
Void main(){

 print('Hello World');
```

}

Kama uta run code hizo utapata majibu haya **Hello World**

```
Hello World

Process finished with exit code 0
```

Sasa somo letu linaanzia hapa.

1. File extension

File extension ya dart ni .dart.

1. Mwanzo wa program

Dart inatumia mfumo kama baadhi ya programming lnaguage nyingine kama java. Kwamba mwanzo wa program ni kwenye function ya **main()**. Hapa ndipo panapoanziwa program ya dart. Kwa kuwa function yetu hai return data ndio maana hapo mwanzo kumeanza na meno **void** kama linavyosomeka hapo **void main(){}**. Hata hivyo unaweza kuondoa neno **void** na bado program yetu ika run vyema.

Ukiangalia vyema hao utaona kuna mabano {} baada ya neno main(). Ipo hivi neno main ni jina la function na ndani ya badno () hapo kutakaa parameter nyingine zo za function na ndani ya mabano {} kutakaa code abazo zinatakiwa zifanye kazi endapo program ita run.

Utaona hapo ndani ya mabano ya {} kuna code zinasomeka print('Hello World');

Hizi ni code ambazo zitatakiwa kufanya kazi wakati program yetu ina run. Sasa hapo kuna function nyingine ya print(). Hii ni kwa ajili ya ku print text kwenye screen. Wacha nikupe mfano mwingine.

```
main(){
 print(3 + 6);
}
```

```
9

Process finished with exit code 0
```

Hapo umeona tumetumia mahesabu sasa tume jumlisha 3 na 6. Pia tumeondoa neno void. Katika mifano yetu hiyo miwili neno **print, main, void** hufahamika kama **keyword**.

1. Matumizi ya **semicolon (;)**

Dart hutumia semicolon ili kutenganisha statement moja na nyingine. Kama ilivyo kwenye lugha nyingine. Angalia mfano huu

main(){

print('Karibu Bongoclass')

}

Hapo uki run hizo code utapata error kwa sababu hakuna semicolon kwenye function ya print. Ilitakiwa iwe print('Karibu Bongoclass');

```
D:/dart-sdk/bin/dart.exe --enable-asserts "C:\User
lesson1.dart:2:12: Error: Expected ';' after this.
 print(args)
           ^
```

Hivyo basi ni muhimu kuzingatia matumizi ya semicolon kwenye Dart kama ilivyo kwenye lugha nyingine.

1. Kuacha nafasi wazi:

Dart haiangalii mistari ilioachwa wazi. Unaweza kuruka nafasi kadiri unavyotaka hakuna tatizo. Program yenyewe itakwenda kuangalia code na sio nafasi. Dart inazingatia bnafasi moja tu ili kuppisha keyword na keyword nyingine. Kwa mfano hapo kama hutaacha nafasi kutoka kwenye neno void na neno main program yako itakuwa na shida

voidmain(){

print('Karibu Bongoclass');

}

Hiyo program hapo haiwezi ku run kwa kuwa ina matatizo.

1. comment

Ili kuweka comment dart inatumia alama ya backslash mbili yaani (//) kwa ajili ya comment ya mstari mmoja. Na kama comment yako ni paragraph yaani mistari zaidi ya mmoja utatumia /* */.

Comment hutumuka kuweka kumbukumbu kwenye code ama kutoa maelekezo kwa watakaokuja kutumia hizo code. Comment hazihusiki kwenye kufanya program ifanye kazi. Comment hazitokei kwenye screen.

Mfano wa comment za single line

//mwanzo wa program

void main(){

//print text kwenye screen

print('Karibu Bongoclass');

}

```
Karibu Bongoclass

Process finished with exit code 0
```

Mfano wa comment za mistari zaidi ya mmoja

//mwanzo wa program

void main(){

*/*huu ni mfano wa*

comment ambazo zina mistari

zaidi ya mmoja

**/*

print('Karibu Bongoclass');

}

```
Karibu Bongoclass

Process finished with exit code 0
```

Pia kuan aina nyingine za comment ambazo hutumia ///. Hizi utumika kwenye kufanya documentation ya program. Mara nyingi hizi hutumika kama unatengeneza kibrary.

7. Case sensitivity

Dart ni case sensitivity kwa maana kuwa husingatia matumizi ya herufi ndogo na kubwa. Kwa mfano neno **Print** na neno **print** ni tofauti. Kwa sababu hilo la kwanza limeanzwa na herufi kubwa na hilo la pili ni ndogo.

Mfano

void main(){

 Print('Karibu Bongoclass');

}

Hizo code hapa zitakupa error kwa sababu nimetumoia P badala ya p

```
Print('Karibu Bongoclass');
^^^^^
```

Mwisho

Katika somo linalofuata tutakwenda kujifunza jinsi ya kutengeneza variable na kuziwekea value zake.

...**Mkala hii imeandaliwa Bongoclass. Kupata makala zaidi tembelea website yetu www.bongoclass.com**...

DART SOMO LA 3: AINA ZA DATA

Summary: Katika somo hili utakwenda kujifunza aina za data zinazotumika kwenye Dart.

Dart ni kama lugha nyinginezo za kompyuta, karibia aina za data zinazotumuka kwenye dart ni sawa katika language nyinginezo. Aina hizo za data ni :-

1. Namba
2. String
3. Boolean
4. Lists
5. Map
6. Runes

1. Namba ni aina za data ambazo ni namba tu.

Namaba zinaweza kuwa ni int yaani namba nzima ama double yaani namba zenye desimali.

void main(){

 print(2);

 print(1.5);

}

```
2
1.5

Process finished with exit code 0
```

1. String

Hizi ni data zilizo katika mfumo wa mchanganyiko wa herufi, namba, na alama za uandishi. String huandikwa ndani ya alama za funga na fungua semi mbili (**“)** ama moja (**‘**). Jambo la kuzingatia ni kuwa utakayoa anza nayo ndio umalizie nayo. Yaani ukitumia (“) mwanzo basi na mwisho utumie (“)

void main(){

print("bongoclass");

print('info@example.com');

print("#@?");

}

```
bongoclass
info@example.com
#@💛

Process finished with exit code (
```

Pia unaweza kuunganisha string zaidi ya moja yaani kufanya concatnation. Kuna njia 3 ambazo hutumika kuunganisa string zaidi ya moja. Njia hizo ni kama

1. Kwa kutumia alama ya kujumlisha +
2. Kwa kutenganisha string mbili kwa kuacha nafasi kati yao
3. Kwa kutumia interpolation yaani unaunganisha string na variable.

void main(){

//concatnation kwa kutumia +

print("karibu "+"Bongoclass");

//kwa kucha nafasi kati ya string mbili

print('karibu ' 'Bongoclass');

//kufanya interpolation

//x ni variable. Tutajifunza zaidi kwenye somo lijalo

var x = 'bongoclass';

print('karibu ${x}');

}

```
karibu Bongoclass
karibu Bongoclass
karibu bongoclass

Process finished with exit code 0
```

Kwenye string unaweza ku break line kwa kutumia back slash () ikifuatiwa na herufi **n** hii ni sawa na matumizi ya <br> kwenye html

```
void main(){

print(' karibu bongoclass upate kujifunza');

}
```

```
karibu
bongoclass
upate
kujifunza

Process finished with exit code 0
```

Kuna changamoto nyingine angalia mfano huu

```
void main(){

print("mama anauza ng'ombe");

}
```

```
mama anauza ng'ombe

Process finished with exit code 0
```

Hapa utaona hakuna error. Ila sasa wacha tutumia single quotation mark (‘)

```
void main(){

print('mama anauza ng'ombe');

}
```

```
lesson1.dart:2:28: Error: Expected ';' after this.
print('mama anauza ng'ombe');
                        ^^^
```

Hapo kwenye neno ng’ombe hiyo alama ya apostroph (‘) haiwezi kuendana na hizo funga semi. Hivyo kutatua tatizo hilo tutatumia back slash 2WD ().

void main(){

print('mama anauza ng'ombe');

}

```
mama anauza ng'ombe

Process finished with exit code 0
```

3. List

Hizi ni data zilizo katika orodha ya vitu vingi. List huandikwa kwenye mabano [ ]. Tunaweza kuitumia kila data iliyo ndani ya list kivyake kwa kutumia indexing. Tutakuja kujifunza zaidi kuhusu undexing.

void main(){

print([1,2,3,4,5]);

print(['bongoclass', 'google', 'facebook']);

var x = ['Tanzania', 'Kenya', 'Uganda'];

print(x);

print(x[1]);

}

```
[1, 2, 3, 4, 5]
[bongoclass, google, facebook]
[Tanzania, Kenya, Uganda]
Kenya

Process finished with exit code 0
```

4. Boolean

Hizi ni data zenye majibu mawili tu yaani True na False. Ni sawa na kusema On na off.

Mfano 2 ni kubwa kuliko 1 hivyo jibu ni kweli.

void main(){

print(2 >1);

}

```
true

Process finished with exit code 0
```

5. Map

Data hizi zinafanana kidogo na list. Tofauti ni kuwa zenyewe zipo kama na pea kuna key na value. Ila key haitakiwi kufanana na nyingine. Data hizi zinakuwa kwenye mabano haya {} tofauti na list ambazo zilitumia mabani haya ().

Ili uweze kuandika map utaanz ana mabano {} kisha utaweka key ya hiyodata. Kama ni string itakaa ndani ya zile alama za finga na funguwa semi. Kisha itafuatiwa na nukta pacha (:) kisha utaandika value yaani thamani ya hiyo data. Kutenganisha itemmoja na nyingine utatumia alama ya koma (,)

Unapotaka kuitumia item yeyote kwenye list tutatumia key yake. Kwa mfano angalia jinsi nilivyo display jina la juma kmwey mfano hapo chini.

void main(){

print({'jina':'Bongoclass','umri':6, "status":'live' });

var x = {'jina': 'juma', 'umri': 35, 'uraia': 'Tanzania'};

print(x['jina']);

}

```
{jina: Bongoclass, umri: 6, status: live}
juma

Process finished with exit code 0
```

6. Runes

Hii hutumika pale unapotaka ku display symbol kwa mfano zile emoj. Hapa tutatumia alama ya back slash.

void main(){

print('u2665');

print('u{1f600}');

}

```
♥
😀

Process finished with exit code 0
```

Jinsi ya kujuwa aina ya data iliyotumika

Kufanya hivi tutatumia runtimeType Angalia mfano hapo chini.

```
void main() {

 print(2.runtimeType);

 print("haloo".runtimeType);

 print(2.9.runtimeType);

 print(true.runtimeType);

 print([2,'tatu', 4].runtimeType);

 print({'jina':'bongoclass'}.runtimeType);

}
```

```
int
String
double
bool
List<Object>
_Map<String, String>
```

Mwisho

Tukutane somo linalofuata titakwenda kujifunza kuhusu variable.

...**Mkala hii imeandaliwa Bongoclass. Kupata makala zaidi tembelea website yetu www.bongoclass.com**...

DART SOMO LA 4: JINSI YA KUANDIKA NA KUTUMIA VARIABLE KWENYE DART

Summary: Katika somo lililopita tumejifunza kuhusu sintaksia za dart. Katika somo hili utakwenda kujifunza jinsi ya kutengeneza variable kwenye Dart.

Nini maana ya variable?

Variable ni jina ambalo hutumika kuhifadhia data kwenye program. Jina la variable huitwa **identifier.** Kwa mfano ukisema x = 6 yaani x ni sawa na 6. Hivyo tunasema x ni variable ambayo inahifadhi 6

Sheria za variable:

1. Haitakiwi kuanza na special character
2. Haitakiwi kuanza na namba
3. Haitakiwi kuanza na underscore
4. Variable ni case sensitivity kwa maana unatakiwa uzingatie matumizi ya herufi ndogo na kubwa. Mfano Juma na juma ni majina mawili tofauti.

Jinsi ya kuandika variable:

Kwanza utaanza na jina la variable (**identifier)** ikifuatiwa na alama ya (=) ikifuatiwa na value ya hiyo variable. Wakati mwingine utaanza na kutaja aina ya data ikifuatiwa na jina la vaiable likifuatiwa na alama ya (=) ikifuatiwa na value

Kwenye Dart kuna njia njingi za kuandika variable kuliko baadhi ya luga nyinginezo:-

1. Kwa kutumia keyword **var**
2. Kwa kutumia **dynamic**
3. Kwa kutumia **const**
4. Kwa **final**

1. Utaanza na neno **var** likifuatiwa na jina la variable. Kisha utaweka alama ya = iikifuatiwa na value (thamani) ya variable. Unapotumia **var** dart yenyewe itaelewa aina ya data aina ya data itakayotumika.

Mfano

```
void main(){

 var web = 'bongoclass';

 print(web);

}
```

```
bongoclass

Process finished with exit code 0
```

Wakati mwingine kwa sababu za ki usalama utahitaji aina maalumu ya data kwenye variable yako. Hapa utataja aina ya data moja kwa moja

void main() {

String web = 'bongoclass';

print(web);

}

```
bongoclass

Process finished with exit code 0
```

Kama ukiweka aina nyingine iliyo tofauti, na ambayo umeitaja itakuletea error. Angalia mfano hapo chini

import 'dart:ffi';

void main() {

Int web = 'bongoclass';

print(web);

}

```
- 'Int' is from 'dart:ffi'.
Int web = 'bongoclass';
          ^
```

Hapo kuna error kwa sababu aina ya data nilio target ni int yaani namba lakini value nilioweka ni string.

Pia unaweza kutengeneza variable zaidi ya moja kwenye mstari mmoja kwa kutumia aina ya data moja. Angalia mfano hapo chini variable a, b na c ni int yaani namba. Zote nimezitengeneza kwa pamoja.

void main() {

int a, b, c;

a = 5;

b = 10;

c = 2;

print(a +b +c);

}

Hapo jibu ni 17

```
17

Process finished with exit code 0
```

Pia unaweza kubadili value ya variable kwa kuitumia tena. Ila hakikisha type of data inabakia ile ile kama umetumia var.

```
void main() {

var x = 3;

x = 4;


print(x);

}
```

Wakati mwingine unahitaji variable iweze kutumika zaidi ya mara moja kwa value tofauti tofauti na aina tofauti tofauti za data Hapa tutatumia **dynamic**. Tofauti iliyopo kati ya dynamic na var ni kuwa unapotumia jina la variable kwenye **var** huhusiwi kubadili aina ya data. Ila unapotumia **dynamic** hilo jina linaweza kutumiwa tena na kubadili value.

```
void main() {

 dynamic x = 'haloo';

 print(x);

 x = ['hi', 'hujambo'];

 print(x[1]);

 x = {'salamu':'habari'};

 print(x);

}
```

```
haloo
hujambo
{salamu: habari}

Process finished with exit code 0
```

```
haloo
hi
habari

Process finished with exit code 0
```

Mfano kama huo ukitumia **var** hauwezi kuleta matokei hayo.

```
void main() {
 var x = 'haloo';
 print(x);
 x = ['hi', 'hujambo'];
 print(x[1]);
 x = {'salamu':'habari'};
 print(x);
}
```

```
 - 'List' is from 'dart:core'.
  x = ['hi', 'hujambo'];
      ^
lesson1.dart:6:7: Error: A value
 - 'Map' is from 'dart:core'.
  x = {'salamu':'habari'};
      ^
```

Final na **const**

Hizi ni keyword nyingine ambazo utumika katika kutengenezea (declare) constant. Contant ni kama variable zilivyo ila utofauti wake mkubwa ni kuwa constst haiwezi kubadilika. Yaani unapotumia jina kuwakilisha value huwezi kulitumia tena jina hilo. Yaani jina la hiyo variable haliwezi kutumika tena.

```
void main() {
 const web = 'bongoclass';
```

```
 final site = 'google';

 print(web);

 print(site);

}
```

```
bongoclass
google

Process finished with exit code 0
```

Huwezi tena kufanya kama ambavyo tulifanya kwenye var na dynamic. Itakupa error.

```
void main() {

 const web = 'bongoclass';

 web = 'youtube';

 final site = 'google';

 site = 'microsoft';

 print(web);

 print(site);

}
```

Hapo utagundua kuwa final na const maana zao zinakaribiana sana. Ila utofauti ni kuwa const value yake una uhakika nayo kuwa haitabadilika wakati wa ku run code. Kwa mfano unajuwa kuwa unga kipishori kimoja ni kilo 25. Una uhakika na vlue na unaijuwa hapo tutatumia const.

Ila kama hauijui value hapa utatumia final. Kwa mfano kama data zinatoka kwenye API hapa kua uwezekano mkubwa wa data kuwa na mabadiliko muda wowote kwa kuwa huwenda wewe so mmiliki wa hizo data. Ama mfano mzuri ni muda, unaweza kubadilika kila wakati. Hivyo hapa utatumia final.

Mwisho:

Somo linalofuata tutakwenda kujifunza kuhusu Dart keywords

...**Mkala hii imeandaliwa Bongoclass. Kupata makala zaidi tembelea website yetu www.bongoclass.com**...

DART SOMO LA 5: RESERVED KEYWORDS KWENYE LUGAH YA DART

Summary: Katika soo hili utakwenda kujifunza kuhusu reserved keywords katika Dart

Keyword ni nini kwenye programming language?

Haya ni maneno ambayo yana maana maalumu kwenye lugha hiyo, hivyo hayawezi kutumika kama identifier kutumika kama jina la variable, class, object ama function.

Hapa nyimi nitakuletea orodha ya hizi keywords

Keywords Table					
abstract	else	import	super	as	enum
in	switch	assert	export	interface	sync
async	extends	is	this	await	extension
library	throw	break	external	mixin	true
case	factory	new	try	class	final
catch	false	null	typedef	on	var
const	finally	operator	void	continue	for
part	while	covariant	Function	rethrow	with
default	get	return	yield	deferred	hide
set	do	if	show	dynamic	implements
static					

Mwiisho:

Tutakwenda kujifunza matumizi ya hizi keyword kila tunapoendelea na somo letu. Kwa mfano mpaka kufikia hapa kuna baadhi ya keywords tulisha zitumia kama **var, const, dynamic, return** na nyinginezo.

...**Mkala hii imeandaliwa Bongoclass. Kupata makala zaidi tembelea website yetu www.bongoclass.com**...

DART SOMO LA 6: DART OPERATOR NA JINSI ZINAVYOFANYA KAZI.

Summary: Katika somo hili utakwenda kujifunza kuhusu operator kwenye Dart, kazi zake na jinsi ya kuzitumia.

Operator ni nini?

Operator hizi ni aalama ambazo hutumikakwa ajili ya kubadili value ama kufanya mahesabu mbalimbali. Ukisha elewa maana ya operator itakubidi uelewe maana ya operand. Sasa ili uelewe vyema viwili hivi nitakupa mfano:

2 + 3 = 5 katika mfano huu namda 2 na 3 ni **operand** na namba alama ya kujumlisha (+) ni operator. Operator zimegawanyika katika makundi mbalimbali ambayo ni

1. Arithmetic operator
2. Assignment operator
3. Relational operator
4. Type test operator
5. Logical operator
6. Bitwise operator
7. Conditional operator
8. Casecade notational operator

Katika somo hili tutakwenda kuiangalia aina moja ya operator ambayo ni arithmetic operator.

ARITHMETIC OPERATOR

Operator hizi ni zile ambazo tumezoea katika matendo ya hesabu. Ambazo ni

1. Addition (+) Hitumika kujiumlisha mfano **10 +4 = 14**
2. Substruction (-) Hutumika kutoa mfano **10 - 4 = 6**
3. Divide (/) hutumika kugawanya mfano **10 / 5 = 2**
4. Multiplication (*) hii hutumika kuzidisha mfano **3 * 6 = 18**
5. Modulus (%) hii ni sawa na kugawanya ila badala ya kutoa jibu itatoa namba iliyobaki. Mfano 10 % 3 jibu ni 1. Hii inakupa namba iliyobaki.

6. Division (~/) hii hutumika kugawanya ila yenyewe itakupa jibu tu pila ya kuleta desimali kama kuna kilichobaki. Mfano 10 ~/3 hapa jibu ni 3. Bila ya kuleta desiali
7. Urinary minus (**-()**). Hii hutumika kubadili alama iliyopo kwenye operand. Kama ni negative inakuwa positive na kama ni positive inakuwa negative. Mfano **- (10 - 4) = -6** utaona hapo jibu ni **- 6** kwa kuwa tumebadli 10 kutoka positive kuwa negative. Kisha tukatoa negative 10 kutoa negative 4 jibu negative 6.

Agalia mfano hapo chini

```
void main() {

 print(5 + 6);

 print(7 - 5);

 print(10 / 3);

 print(10 ~/ 3);

 print(10 % 3);

 print(-8 - 4);

 print(3 * 4);

}
```

```
11
2
3.3333333333333335
3
1
-12
12

Process finished with exit code 0
```

URINARY OPERATOR

Hizi ni operator ambazo zinafanya kazi ya incrementon (**++**) na decrementation (**– -**) kwenye value. Kufanya increment ni kuongeza namba moja kwenye value. Mfano ++20 inakuwa 21. Na decrement ni kutoa namba moja kwenye value. Mfano - - 20 inakuwa 19.

Sasa kwenye Dart kuna post na pre. Unaposema post maana yake hizi alama zinakaa mbele ya value na kazi yake ni kutuma namba halisi kabla ya increment ama decrement mfano **20++** hapa itakupa jibu 20 kwa kuwa

ndio namba halisi kabla ya increment. Na **10 - -** itakupa jibu 10 kwa kuwa ndio namba halisi kabla ya decrement.

```
void main() {
 var a, b, c, d;
 a = 10;
 b= 20;
 c = 30;
 d = 14;

 print(a++);
 print(++b);
 print(c--);
 print(--d);
}
```

```
10
21
30
13

Process finished with exit code 0
```

ASSIGNMENT OPERATOR

Hizi ni operator ambazo hutumika kwa ajili ya kuipa variable thamani yaani value. Hapo awali tuisha jifunza alama ya (=) katika kuipa variable value yake. Sasa hapa tutakwend akuchanganya na operator nyingine ili kuweza ku assign value kwenye variable:-

Miongoni kwa assignment opetaror ni kama :-

1. Assignment operator (=) hiitumesha izungumzia
2. Add and assign (+=) hii kari yake ni kuongeza value operand ya kulia kwenye value ya operand ya kushoto. Mfano a = 3 na b = 4 sasa kama nitaandika hivi **a += b** hap **a** ndio operand ya kushoto na **b** ndio operand ya kulia ina maana sawa na kuandika hivi **a = b + a**. Ambapo sasa **a** itapata value mpya ambayo ni 7.

3. Substract and assign (- =). Hii yenyewe itatoa value kwenye operant ya kushto na kurudisha value kwenye operand ya kushoto. Mfano a 10, b = 4 hivyo nikiandika **x - = y** ni sawana kuandiak **x = x - y** hivyo value mpya ya variable **x** ni 6
4. Multiplay and assign (* =) mfano k = 10, j = 2 hivyo nikiandika **k * = j** ni sawa na kuandiaka **k = k *j** hivyovalue mpya ya variable **k** itakuwa **20**
5. Divide and assign (/=) mfano c = 8, d = 4 hivyo **c/=d** ni sawa na kuandika **c = c/d** abapo value mpya ya variable **d i**takuwa 2.
6. Divide and assign (~/=) **e = 10, f = 3** hivyo **e ~/=f** ni sawa na kuandiiak **e = e~/f** ambapo value mpya ya variable **e** itakuwa ni 3
7. Mod and assign (%=) **g = 10, h = 3** hivyo **g %=f** ni sawa na kuandika **g = g %f** ambapo value mpya ya variable **g** ni 1

```
void main() {

 var a = 4;

 var   b = 3;

 a +=b;

 print(a);


 var x = 10;

 var y = 4;

 x -=y;

 print(x);


 var k = 10;

 var j = 2;

 k *=j;

 print(k);



 dynamic c = 8;

 var d = 4;

 c /=d;

 print(c);
```

```
var e = 10;

var f = 3;

e ~/=f;

print(e);


var g = 10;

var h = 3;

g %=h;

print(g);


}
```

```
7
6
20
2.0
3
1

Process finished with exit code 0
```

RELATIONAL OPERATOR

Hizi ni operator ambazo zinaonyesha mahusiano yaliyopo kwenye **item** katia program. Hizi zinatupa **true** ama **false.** Operator hizi zinaweza kuwa:-

1. Kubwa kuliko (>) mfano 5 > 4 ina maana 5 ni kubw akuliko 4. Hapa ni **true**
2. Ndogo kuliko (<) mfano 5 < 4 ina maana 4 ni ndogo kuliko 4. Hapa ni **false**
3. Kubwa kuliko ama sawasawa (>=) mfano 5 >= 4 hii ina maana 5 ni kubwa kulio 4 au ni sawasawa. Hapa ni **true**
4. Ndogo kuliko ama sawasawa (<=) mfano 5 <= 4 hii ni **false**
5. Sawasawa (==) hii huonesha kuwa item 2 zipo sawasawa. Mfano 5 == 5 hii ina 5 ni sawasawa na 5 hii ni **true**
6. Si sawasawa (!=) mfano 5 != 5 hii ina maana 5 sio sawasawa na 5 hii **false**

```
void main() {
```

```
 print(5 > 4);
 print(5 < 4);
 print(5 >= 5);
 print(5 <= 5);
 print(5 == 5);
 print(5 != 5);
}
```

```
true
false
true
true
true
false

Process finished with exit code 0
```

LOGICAL OPERATOR

Hizi hutumika kwenye bool data type.

1. AND (&&)
2. OR (||)
3. NOT (!)

```
void main() {
 print(true || false);
 print(true && false);
 print(false || true);
 print(false && true);
 print(true && true);
 print(false && false);
}
```

```
true
false
true
false
true
false

Process finished with exit code 0
```

Operator nyinginezo tutazisoma kidogo kidogo kadiri somo linavyoendelea.

Mwisho:

Katika somo linalofuata tutakwenda kujifunzakuhusu **control flow** tutakwend akujifunz akuhusu kuweka mashart kwenye program.

...**Mkala hii imeandaliwa Bongoclass. Kupata makala zaidi tembelea website yetu www.bongoclass.com**...

DART SOMO LA 7: MATUMIZI YA IF, ELSE, IF ELSE, ELSE IF KWENYE DART

Summary: Katika somo hili utakwenda kujifunza kuhusu namna ambavyo program inaweza kufanya maamuzi kulingana na mashart fulani.ama namna ambavyo program inaweza ku run code zaii ya mara moja kulingana na mashart. Hapa tutazingumzia flow of control statement

Ili uweze kwanza kuelewa somo hili nitakupa case hapa mbili.

Case 1: umetkiwa ktengeneza program kwa ajili ya menyu ya chakula chuleni. Unatakiwa program hiyo inapofika muda wa kula asubuhi saa 4 iseme **chakula cha asubuhi kipo tayari** na wakati wa mchana saa 7 iseme **chakula cha mchana kipo tayari** na inapofika jioni saa 1 usiku iseme **chakula cha usiku kipo tayari** mwisho inapofika saa 4 usiku iseme **mlale salama**.

Kwa hiyo hapo utaona tumeweka masharti kulingana na saa. Shari la kwanza ifike saa 4 asubuhi ndipo iseme chakula cha subuhi kipo tayari… na kuendelea masharti mengine. Hii ndio tunaita **descision making**

Case 2: umetakiwa kutengeneza table (zile tabo za kuzidisha katika somo la hesabu) ya 2 kwa ajili ya wanafunzi wa darasa la tatu. Sasa badala ya kuandika moja moja utahitaji program yako iandike yenyewe. Hapa

ndio tunaita **loop** .mfano mzui wa loop ni music player, kwamba unawez akuchaguwa mziki ujirudie rudie mara ngapii.

Flow statemenet zimegawanyika katika makundi makuu matatu ambayo ni:-

1. **Descision making:** hapa program inafanya maamuzi kulingana na masharti atakayoweka. Hizi ni kama
    1. **If**
    2. **Ifelse**
    3. **If else if**
    4. **Switch case**
2. **Looping:** hapa program inarudia rudia ku run code zaidi ya mara moja kulingana na masharti yatakayowekwa.. Hizi ni kama :-
    1. **For loop**
    2. **For in loop**
    3. **While loop**
    4. **Do while loop**
3. **Jumping**: hapa program ita run kisha itawacha baadhi ya block za code na kuendelea tena kulingana na masharti yatakayowekwa.

1. **If statement:**

Hii ni sawa na neno la kiswahiki ikiwa. Kwa mfano tunasema ikiwa umri wa mtoto ni miaka 7 program iseme aende shule. Sawa ngoja tuone jinsi ya kufanya program yetu itumie sharti hili.

Jinsi ya kuandika **descision making statemen**

If (weka sharti) {

Weka code

}

Kama unavyoona hapo kwanza utaanza na keyword **if** likifuatiwa na na mabano **()** ndani ya mabano hayo weka charti lako. Ambapo ni umri wa mtoto iwe miaka 7. Kisha utaweka mabano **{}** ambapo ndani ya mabani hayo utaweka nini kitokee kama sharti limefikiwa. Kwa mujibu wa mfano wetu hap tunataka program iseme **aende shule**. Sasa wacha tuone jinsi ya kuandika hizo code.

Kwanza tutaanz akuweka variable yetu ambayoiabeba umri wa mtoto. Kisha tutaweka charti letu **.**

```
void main() {

 var umri = 7;

 if(umri == 7){

   print('aende shule');

 }
```

}

```
aende shule

Process finished with exit code 0
```

Sasa tunataka kusema ikiwa umri wa mtoto ni zaidi ama sawasawa na miaka 7 iseme aende shule.

```
void main() {
 var umri = 7;
 if(umri >= 7){
   print('aende shule');
 }
}
```

```
aende shule

Process finished with exit code 0
```

Unaweza kutumia naoperator nyingine kama <, 1=, <= na nyinginezo kufanya mazoezi zadi.

Sasa hapo kuna tatizo kama umriutakuwa ni chini ya saba hakuna chochote program itasema . hapa sasa tutahitaji **else** satatement, ambayo itaangalia ikiwa umri ni miaka 7 iseme aende shule lakini ikiwa ni tofauti na miaka 7 iseme hajafikia umri. Kufanya hivi tutafuata formla hii

If (sharti) {

Code

}else{

Code

}

Ukiangalia vyeama hapo utaona kuna keyword ya esle imeongezeka. Hii ndio ambayo itatoa nini kifanyike kama sharti halikufikiwa.

```
void main() {
 var umri = 3;
 if(umri == 7){
```

```
  print('aende shule');
 }else{
  print('hajafikia umri');
 }
}
```

```
hajafikia umri

Process finished with exit code 0
```

Hapo nimebadili umri sasa nimeweka mika 3. Umeona hapo program yetu imesea hajafikia umri. Ila kuna tatizo lingine hapo kama ukiweka umri mika zaidi ya saba pia itasema hajafikiaumri.

```
void main() {
 var umri = 18;
 if(umri == 7){
  print('aende shule');
 }else{
  print('hajafikia umri');
 }
}
```

```
hajafikia umri

Process finished with exit code 0
```

Utaona hapo nimeweka miaka 18 lakini bado naabiwa hajafikia umri. Kutatua tatizo hili tatatakiwa tuwe na mlolongo mzuri. Kwa mfano tuseme

Ikiwa umri ni chini ya miaka 7 iseme hajafikia umri, na ikiwa ni miaka 7 iseme aende shule. Na ikiwa ni zaidi ya miaka saba iseme umri ni mubwa zaidi.ukiangalia vyema hapo kuna masharti mawili. Sasa kutekeleza hili ndipo tutahitaji kutumia **else if**. Kanuni yake ni hii

if(sharti la 1){

Code

}elseif(shari la 2){

Code

}else{

Code

```
}
void main() {
 var umri = 18;
 if(umri < 7){
   print('hajafikia umri');
 }else if(umri == 7){
   print('aende shule');
 }else{
   print('umri ni mubwa zad');
 }
}
```

```
umri ni mubwa zad

Process finished with exit code 0
```

Unaweza kuongeza masharti mengi zaid bila kikomo. Wacha tuone mfano mwingine. Hapo kwenye program yetu mwenye umri zaidi ya mika 7 inasema anaumri mkubwa zaidi. Sasa tunataka ikiwa umri ni zaidi ya kiaka 7 na ni chni ya miaka 14 basi program iseme ukaanze memkwa, vinginevyo ndipo iseme umri ni mkubwa zaidi. Tutakwenda kutest kwa umri wa miaka 10.

```
void main() {
 var umri = 10;
 if(umri < 7){
   print('hajafikia umri');
 }else if(umri == 7){
   print('aende shule');
 }else if (umri >7 && umri<= 14){
   print('akaanze memkwa');
 }else{
   print('umri ni mubwa zad');
 }
}
```

```
akaanze memkwa

Process finished with exit code 0
```

Endelea kufanyia kazi operator tulizozisoma kwenye haya mashari. Umeona hapo nimetumia operator <=, >, na &&. Jambo la msingi kuzingatia ni kuwa hakuna ukmo wa kutumia **else if** ama **if** yeyewe unawez akuzitumia zaidi na zaidi.

Mwisho:

Katika somo linalofuata tutakwenda kujifunza kuhusu switch case.

...**Mkala hii imeandaliwa Bongoclass. Kupata makala zaidi tembelea website yetu www.bongoclass.com**...

DART SOMO LA 8: MATUMIZI YA SWITCH CASE

Summary: Katika somo hili utakwenda kujifunza kuhusu switch case na jinsi inavyofanya kazi kwenye Dart.

Katika somo hili tutakwenda kutumia mifano yetu ya somo lililotangulia ili kuweza kujienga uelewa zaidi.

Jinsi ya kuandika switch case

switch(expression){

Case value:

Code

break

}

Ukiangalia hapo ni kuwa utaanza na keyword **switch** utaweka mabano () kama kawaid ndani yake ni value tunayotaka ku test ikufuatiwa na mabano {}. Kisha utaweka keyword **case** ikifuatiwa na sharti unayotaka ku test kisha utawka nukta pacha (**:**). Baad aya hapo ni code ambazo zita run kisa utaweka keyword **break** kwa akiji ya ku stop code zisiendelee ku run

Wacha tuone mambo yanavyokuwa:

Tunataka ikiwa umri wa mtoto ni miaka 7 aambiwe aende shule.

```
void main() {
 var umri = 7;
 switch(umri){
  case 7:
    print('aende shule');
    break;
 }
}
```

```
aende shule

Process finished with exit code 0
```

Sasa tunataka kama ni chini ya kiaka 7 aambwe hajafika umri, na kama miaka 7 aambiwe aende shule, na kama ni miaka 13 au zaidi aambiwe aende mmemkwa. Tutakwenda ku test miaka 10

```
void main() {
 var umri = 10;
 switch(umri){
  case <7:
    print('aende chekechea');
    break;

  case ==7:
    print('aende shule');
    break;

  case >=13:
    print('aende memkwa');
    break;
 }
```

```
}
```

Hapo utaona hakuna matokeo yeyote. Ni kwa sababu 10 haopo kwenye test zetu. Kwahiyo hapo tunatakiwa kuweka **default** yaani tuweke code ambazo zitafanya kazi ikiwa sharti hata moja halijafikiwa.

Sasa tunataka ikiwa hakuna sharti lolote lililofikiwa basi iseme umesha **je umesoma?**

```
void main() {
 var umri = 10;
 switch(umri){
   case <7:
     print('aende chekechea');

   case ==7:
     print('aende shule');
     break;

   case >=13:
     print('aende memkwa');
     break;

   default:
     print('je umesoma?');
 }
}
```

Hakuna ukomo wa kutumia switch case unaweza kutumia zaidi na zaidi. Miongoni mwa fgaida zake ni kuwa ni rahisi kuitumia, na inapunguza mrundikano wa code nyingi kuliko kwenye if else else if.

...**Mkala hii imeandaliwa Bongoclass. Kupata makala zaidi tembelea website yetu www.bongoclass.com**...

DART SOMO LA 9: FOR LOOP NA FOR IN LOOP KWENYE DART, KAZI ZAKE NA JINSI YA KUADIKA

Summary: Katika somo hli utakwend akujifunza kuhusu loop kwenye Dart. Loop zimegawanyika katika makundi mnne ambayo ni for loop, while loop, for in loop na do while loop.

For loop

Hii ni moja katika njia ya kurun code mara nyingi zaidi kulingana na masharti husika. **For loop** hutumika kama unaelewa ni mara ngapi code zako zinatakiwa ku run. Tuchukulie mfano tunataka kutengeneza tebo ya 7. Hapa code zetu zita run mara 12.

Jinsi ya kuandika for loop

For (initialization ; condition; increment/decrement){

Code

}

Kama unavyoona hapo kwanza utaanza na keyword **for** ikifuatiwa na mabano ambayo ndani yake utaanza na **initial** huu ni mwnzo ambapo program yako inatakiwa kuanza ku run..

Baada ya initialization itafuata kuweka **condition** yaabu sharti ambalo code zako zitaliangalia. Utatenganisha initialization na condition kwa kutumia alama hii (**;**). Kisha itafuata **increment** au **decrement**. Operator. Kisha utaweka mabano {} ambayo ndani yake ndipo utaweka hizo code zitakazofanya kazi.

```
void main() {

 print('TEBO YA 7');

 for (int x = 1; x <=12; x++){

   print(7*x);

 }

}
```

```
TEBO YA 7
7
14
21
28
35
42
49
56
63
70
77
84
```

Kuna namna zaidi tunaweza kuboresha code zetu. Ni kwa kutumia **${}** hii hutumika kuweza kutumia interpolation (rejea string interpolation kwenye somo la type of data).

```
void main() {

 print('TEBO YA 7');

 for (int x = 1; x <=12; x++){

   print('${x}*7 =${x*7}');

 }

}
```

```
TEBO YA 7
1*7 =7
2*7 =14
3*7 =21
4*7 =28
5*7 =35
6*7 =42
7*7 =49
8*7 =56
9*7 =63
10*7 =70
11*7 =77
12*7 =84
```

For in loop

Hii hutumika kwenye list, object au map. Tofauti na for loop ambayo inawekewa condition. Hii yenyewe inafanya kinachotakiw akwenye list husika.

Kanuni ya kuandika for in loop

```
for (var in expression) {

   code

}
```

Kama unavyoona hapo kwanza utaanza na keyword for ilifuatiwa na mabano () ndani yake utaanza na intializationa kama tulivyoona hapo mwanzo kisha itafuatiwa na keyword in baadaye utaweka hiyo data yako.

```
void main() {

 print('TEBO YA 7');

 var tebo = [7,14,21,28,35,42,49,56,63,70,77,84];

 for(var x in tebo){

   print(x);

 }

}
```

```
TEBO YA 7
7
14
21
28
35
42
49
56
63
70
77
84
```

Sio tu ku print hiyo list bali pia tunaweza kuifanyia mahesabu kama tulivyofanya kwenye for loop.

```
void main() {

 print('TEBO YA 7');

 var tebo = [1,2,3,4,5,6,7,8,9,10,11,12];

 for(var x in tebo){
```

```
  print(x *7);
 }
}
```

```
TEBO YA 7
7
14
21
28
35
42
49
56
63
70
77
84
```

```
void main() {
 print('TEBO YA 7');
 var tebo = [1,2,3,4,5,6,7,8,9,10,11,12];
 for(var x in tebo){
  print('${x} *7= x*7');
 }
}
```

```
TEBO YA 7
1 *7= x*7
2 *7= x*7
3 *7= x*7
4 *7= x*7
5 *7= x*7
6 *7= x*7
7 *7= x*7
8 *7= x*7
9 *7= x*7
10 *7= x*7
11 *7= x*7
12 *7= x*7
```

Mwisho:

Unaweza kufanyia kazi aina nyigine za data, ama kutengeneza mifano zaidi na zaidi. Somo linalofuata tutakwenda kujifunza kuhusu **while loop**.

...**Mkala hii imeandaliwa Bongoclass. Kupata makala zaidi tembelea website yetu www.bongoclass.com**...

DART SOMO LA 10: WHILE LOOP NA DO WHILE LOOP KWENYE DART

Summary: Katika somo hili utakwend akujifunza kuhusu while loop na do while loop zinavyofanya kazi katika Dart.

While loop hii hutumika endapo hujui kama code zako zinatakiw aku run mara ngapi. Yenyewe hii ni rahisi sana kutumia. Unachotakiwa kufanya ni kuweka sharti (condition) kisha zitafuata code.

While (condition){

Code

increment/ decrement

}

```
void main() {
 print('ORODHA YA NAMBA');
 var x = 1;
 print(x);
 while(x <12){
   x++;
   print(x);
 }
}
```

```
ORODHA YA NAMBA
1
2
3
4
5
6
7
8
9
10
11
12
```

Wach tuone jinsi ambavyo tutaweza kutengeneza tebo yetu ya 7 kwa kutumia while loop.

```
void main() {
 print('TEBO YA 7');
 var x = 0;
 while(x <12){
   x++;
   print("${x}*7= ${x *7}");
 }
}
```

```
TEBO YA 7
1*7= 7
2*7= 14
3*7= 21
4*7= 28
5*7= 35
6*7= 42
7*7= 49
8*7= 56
9*7= 63
10*7= 70
11*7= 77
12*7= 84
```

Do while loop

Hii ni sawa na while loop ila utofauti wao ni kuwa do **while loop** yenyewe kwanza itanza na ku run code husika kabla ya kuangalia sharti. Baad ya ku run mara moja ndipo inaangalia sharti zilizowekwa, kama zitatimia itaendelea ku run ila zisipotimia itastop.

Kanuni yake:

```
do {
code
}while(condition);
```

```
void main() {
 var x = 1;
 print('TEBO YA 7');
 do{
   print("${x}*7= ${x *7}");
   x++;
 }while(x<=12);
}
```

```
TEBO YA 7
1*7= 7
2*7= 14
3*7= 21
4*7= 28
5*7= 35
6*7= 42
7*7= 49
8*7= 56
9*7= 63
10*7= 70
11*7= 77
12*7= 84
```

Ukiangaia vyemahapo kwanza tumianza ku print print("${x}*7= ${x *7}");

Kisha ndipo tukafanya increment. Na baadaye tuakaangalia sharti kama zimefikiwa. Sasa ngoja tubadili shari tuone, tunabadili badala ya **x<=12** tunakwenda kuweka **x>12**

```
void main() {

 var x = 1;

 print('TEBO YA 7');

 do{

   print("${x}*7= ${x *7}");

   x++;

 }while(x>12);

}
```

```
TEBO YA 7
1*7= 7

Process finished with exit code 0
```

Ukiangalia hapo utaona ijapokuwa sharti halijafikiwa lakini code zetu zilianga ku run na kutuma matokeo **1*7= 7**.

Mwisho:

Somo linalofuata tutakwenda kujifunza kuhusu break and continue kweny loop. Hivyo basi edelea kufanyia mazoezi kwa mifano zaidi.

...**Mkala hii imeandaliwa Bongoclass. Kupata makala zaidi tembelea website yetu www.bongoclass.com**...

Kitabu hiki kina muendelezo kwa ajili ya mafunzo zaidi. wasiliana nasi kupata kitabu kizima