

Variables	
Create	var name = '...'
Rewrite	name = '...'
+ or -	num ++ -- **// 1
Double -> Int	(num/3).toInt(); or num ~/ 3
if num-num=1 T or F	bool num-num == 1
var.toString()	int? to string
list:	
Var list = ['...', '...'];	Create list
list.add("name");	add name to list
list.remove("name");	remove name from list
list.removeAt(4)	remove element with index 4
name = list[50]	get element with index 50
list[50] = "name"	rewrite index 50 with name
list.addAll(['...', '...', '...'])	add all to list
list.contains('...')	is ... in list?
list.clear()	clear list
list.length	how long is list
list.indexOf('...')	index of element ... is
Set:	
must be unique	

Variables (cont)	
setlist = {"key": "name"}	map
	Set dont have 'add' only 'addAll'
setlist.keys	get all keys
setlist.values	get all values
---.toList()	itable -> list
setlist.containsKey('key')	is key exist?
setlist.containsValue('value')	is value exist?
Var types	
String	Text
Int	Number
Bool	True/False
Double	Number with .
Final ...	creating while compilation
Static ...	creating while writing code
dynamic	anything
Dart	
dart create -t web-simple name	create dart web project
dart compile js path/to/dart -o path/to/js	compile dart to js
void main() {code}	void-return nothing; main function

HTML	
import 'dart.html';	import DOM
import 'dart:async';	import timer
Create variables using html	
ButtonElement btn = querySelector('#my-Button') as ButtonElement;	
connect button with id myButton to the btn variable	
ParagraphElement output = querySelector('#output') as ParagraphElement;	
connect button with id output to the output variable	
DivElement divc = querySelector('#divv') as DivElement;	
connect container with id divv to the divc variable	
FormElement form = querySelector('#my-Form') as FormElement;	
connect form with id myForm to the form variable	
working with html variables	
btn.onClick.listen((MouseEvent e) { code });	
listen the button	
output.text = 'Pressed!';	
Add text to output variable	
output.style.color = 'blue';	



HTML (cont)	
	add style to output element
output.id = 'asd';	
	add id to the object
container.style.display = 'block';	container-name none or block
	show/hide element
ParagraphElement newElem = ParagraphElement();	
TAB ..id = 'newParagraph'	
TAB ..text = 'new text';	
container.append(newElem);	
container - name of container newElem - name of variable	Create paragraph with teg and text
Future.delayed(Duration(seconds: 2), () { code });	
	2sec timer
window.onKeyDown.listen((KeyboardEvent event) { code });	
output.text = 'Key: \${event.code}';	hearing key pressing and print the key
form.onSubmit.listen((Event e) { TAB e.preventDefault();	
TAB InputElement nameInput = querySelector('#name') as InputElement;	
TAB InputElement emailInput = querySelector('#email') as InputElement;	
TAB String name = nameInput.value ?? '';	
TAB String email = emailInput.value ?? '';	
});	
	form
value after ?? uses if value left is null	

Comands	
DateTime.now()	Time now
if/else, switch, while	
if (1 < 2) {code}	if 1 < 2 then make code 'if' cloud have anything that return bool
else if (2 > 1) {code}	
else {code}	
if (1 > 2 && 2 > 1) {code}	execute if 2 statements are true
	or
return;	stop the function
switch	
switch (name)	get variable name
{	
case "name1":	if variable is name1 do:
...	
break;	stop switch execution
default:	if not exist do:
... }	
while	
while (1 < 2)	execute code while ...
{code}	

Cicles/Func	
for (var i = 0; i <= 5; i++) {code}	repeat 5 times
for (final name1 in list) {code}	name1 every time will be next name in list
for (var i = 0; i < list.length; i++) { list[i] = "-text" }	every cicle elements inside list will change

Cicles/Func (cont)	
Functions	
void name	return nothing
int/string/bool name	return int/string
int/string/bool name	must have return ...;
-	-
void name() {code}	create function
name()	call function
void name(int num) \ name(1)	get numbers
Only main func starts automatically	

