

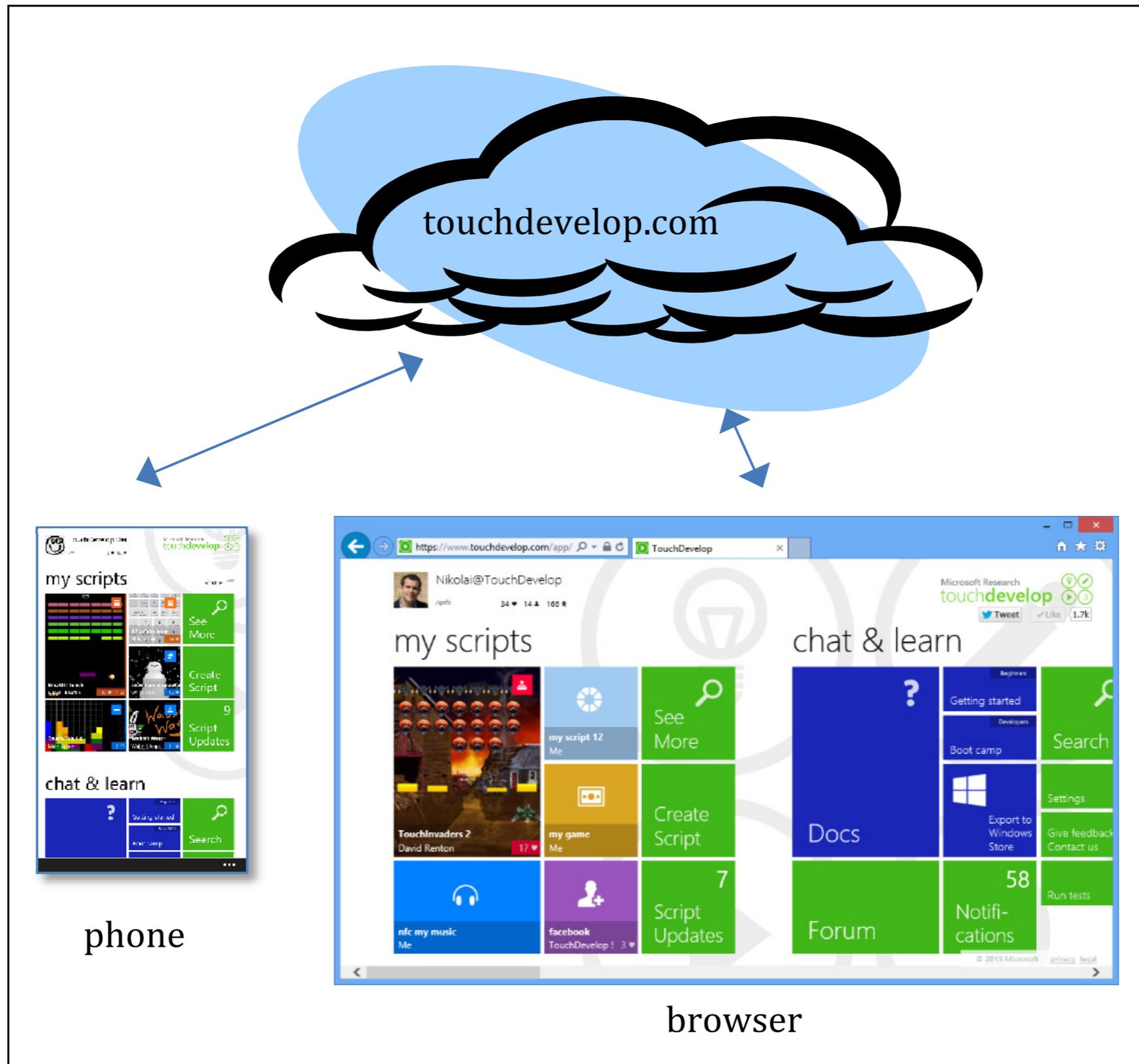
# Scripting with *TouchDevelop*

**<https://www.touchdevelop.com/>**

Introduction to Software Development for Startup  
Entrepreneurs, WHU, Vallendar, March/April 2015

Prof. Dr. Ralf Lämmel  
Software Languages Team  
University of Koblenz-Landau

# The TouchDevelop ecosystem

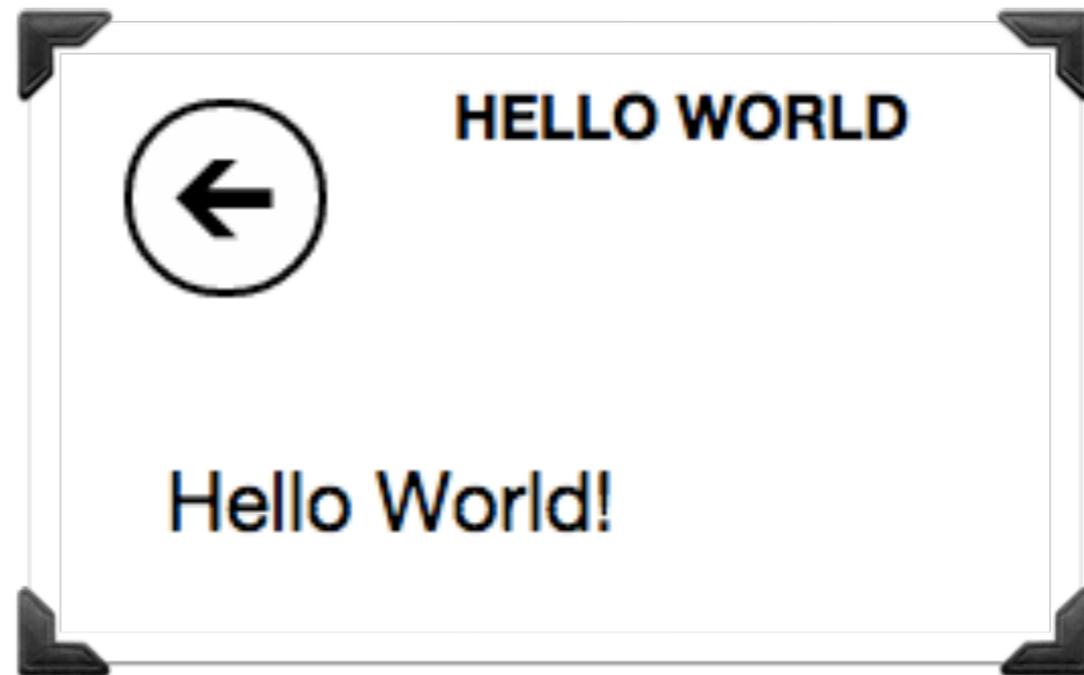


# Hello World

**action** main ()

| "Hello world!" → post to wall

**end** action



# Count from 0 to 9

```
action main ()  
  | for  $0 \leq i < 10$  do  
  |   |  $i \rightarrow$  post to wall  
  | end for  
end action
```

9  
8  
7  
6  
5  
4  
3  
2  
1  
0

# Bark on Shake

```
action main ()  
  | do nothing  
end action
```

```
event shake ()  
  | 🐶puppy bark → play  
end event
```

```
art puppy bark : Sound  
with data: "https://az31353.vo.msecnd.net/pub/eqdbcwzx"
```

# Sharing scripts on Facebook



**Ralf Laemmel**  
Just now

Ok we will be using TouchDevelop to do some programming in the course this year. Run the following script on your phone -- just click the link. Shake the phone. Enjoy the bark.



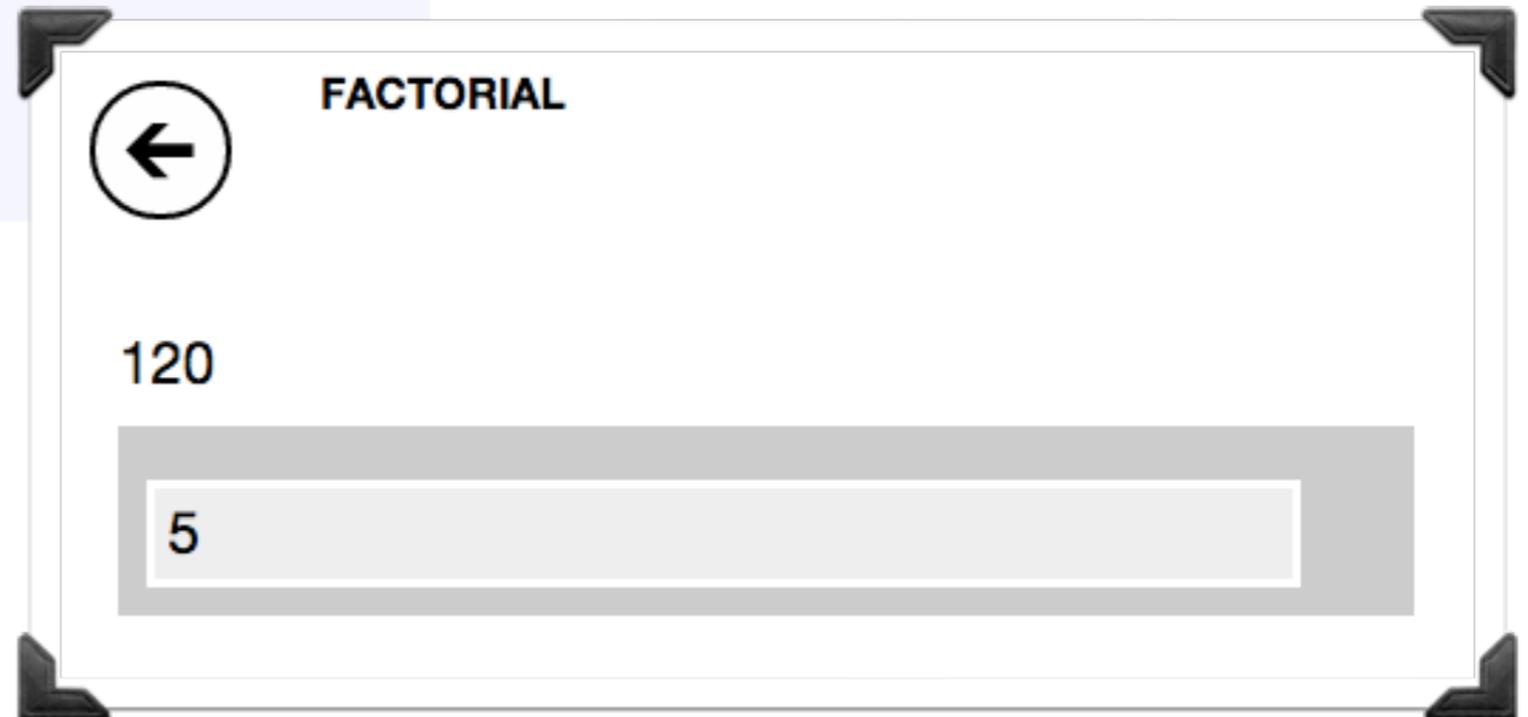
**Bark on Shake by Ralf Laemmel**  
[/yzkgc](#)  
Created with TouchDevelop

TOUCHDEVELOP.COM

Like · Comment · Share

# Factorial

```
action main ()  
  | var arg := wall → ask number("")  
  | var result := 1  
  | while arg > 1 do  
    | result := result * arg  
    | arg := arg - 1  
  | end while  
  | result → post to wall  
end action
```

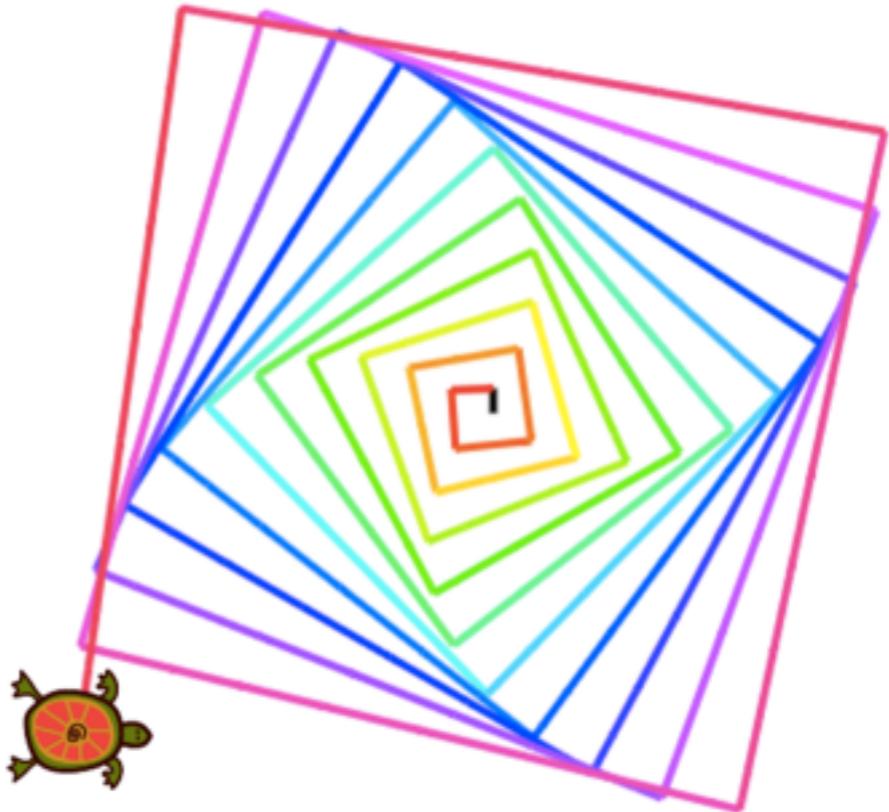


# Spiral Rainbow

my scripts run undo split script

```
action main ()  
  turtle → set speed(2000)  
  for 0 ≤ j < 42 do  
    turtle → forward(10 + j * 7)  
    turtle → left turn(92)  
    var hue := j / 42  
    var rainbow := colors → wheel(hue)  
    turtle → set pen color(rainbow)  
  end for  
end action
```

SPIRAL RAINBOW



# Spiral Rainbow

**action** main ()

⌚ ♻️ turtle → set speed(2000)

**for**  $0 \leq j < 42$  **do**

⌚ ♻️ turtle → forward( $10 + j * 7$ )

⌚ ♻️ turtle → left turn(92)

**var** hue :=  $j / 42$

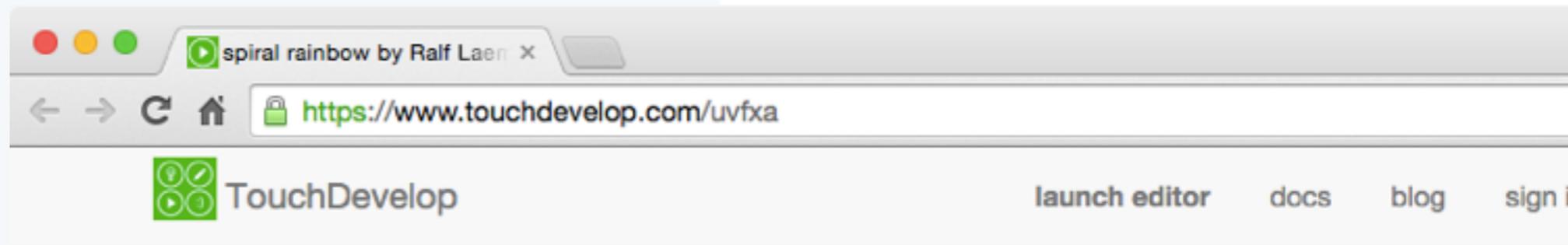
**var** rainbow := colors → wheel(hue)

⌚ ♻️ turtle → set pen color(rainbow)

**end for**

**end action**

# Sharing scripts on Twitter



**spiral rainbow**

**by Ralf Laemmel**

#HourOfCode

run

# A textbox

```
action main ()  
  | var Name := "Ralf"  
  | var tb := wall → create text box("Name = " || Name , 19)  
  | tb → set background(colors → orange)  
  | tb → post to wall  
end action
```



# Displaying a sound

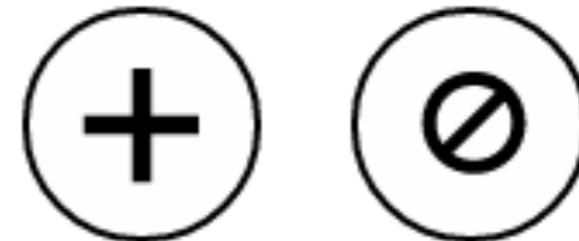
```
action main ()  
  | 🐶puppy bark → post to wall  
end action
```



# Actions et al.

```
action main ()  
  | tb := wall → create text box("", 20)  
  | tb → post to wall  
  | inc := wall → add button("add", " ")  
  | reset := wall → add button("delete", " ")  
  | ▷ update()  
end action  
  
action update ()  
  | tb → set text("Counter = " | count → to string)  
end action  
  
event tap wall Page Button (  
  | item : Page Button)  
do  
  | if item → equals(inc) then  
    | count := count + 1  
  | else  
    | count := 0  
  | end if  
  | ▷ update()  
end event  
  
data count : Number  
  
data inc : Page Button  
  
data reset : Page Button  
  
data tb : TextBox
```

Counter = 6



# An updatable textbox (/censaair)

```
action main ()  
| tbt := wall → create text box("Tap the plus button below", 20)  
| tbt → set border(colors → blue)  
| tbt → post to wall  
| wall → add button("add", "Tap Here")  
end action
```

```
event tap wall Page Button (  
| item : Page Button)  
do  
| tbt → set text("I have been tapped!")  
| tbt → set foreground(colors → red)  
end event
```

```
data tb : TextBox
```

Tap the plus button below



Tap here

# An updatable textbox on a board

```
action main ()
```

```
| board := media → create board(200)
```

```
| sprite := board → create text(200, 20, 30, "Tap the plus button")
```

```
| sprite → set pos(100, 10)
```

```
| sprite → set color(colors → blue)
```

```
| board → post to wall
```

```
| wall → add button("add", "Tap here")
```

```
end action
```

```
event tap wall Page Button (
```

```
| item : Page Button)
```

```
do
```

```
| sprite → set text("I have been tapped")
```

```
| sprite → set color(colors → red)
```

```
| board → update on wall
```

```
end event
```

```
data board : Board
```

```
data sprite : Sprite
```

Tap the plus button



Tap here

# Tapping strings on the wall

```
action main ()
```

```
| "One" → post to wall
```

```
| "Two" → post to wall
```

```
| "Three" → post to wall
```

```
end action
```

```
event tap wall String (
```

```
| item : String)
```

```
do
```

```
| ("\" | item | "\" | " was tapped") → post to wall
```

```
end event
```

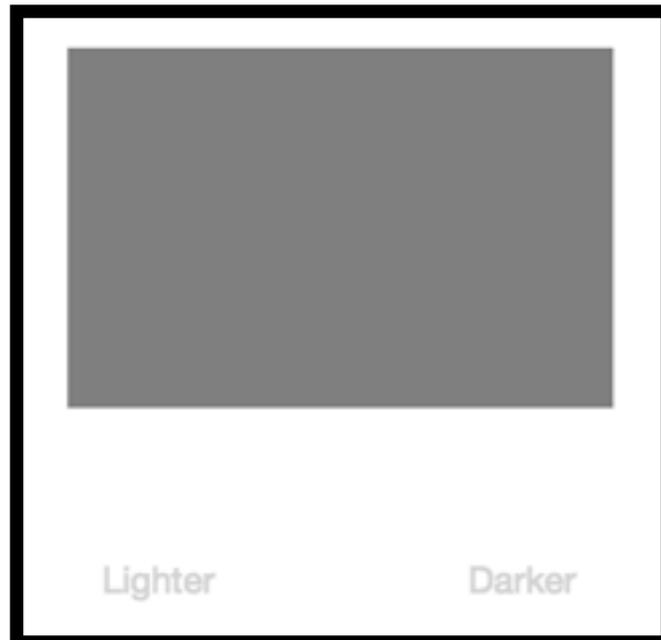
```
"" "Three" was tapped" was tapped" was tapped  
"" "Three" was tapped" was tapped  
"Three" was tapped  
Three  
Two  
One
```

# Sprite events (\akmcnpux)

**action** main ()

```
| board := media → create board(640)
| rectangle := board → create rectangle(300, 200)
| rectangle → set color(colors → from rgb(0.5, 0.5, 0.5))
| rectangle → set pos(200, 200)
| lighter := board → create text(100, 20, 40, "Lighter")
| darker := board → create text(100, 20, 40, "Darker")
| lighter → set color(colors → foreground)
| darker → set color(colors → foreground)
| lighter → set pos(100, 400)
| darker → set pos(300, 400)
| controls := board → create sprite set
| controls → add(lighter)
| controls → add(darker)
| board → post to wall
```

**end action**



**event** tap sprite in controls (

```
| sprite : Sprite,
| index in set : Number,
| x : Number,
| y : Number)
do
| var delta := 0.2
| if index in set = 0 then
| | rectangle → set color(rectangle → color → lighten(delta))
| else
| | rectangle → set color(rectangle → color → darken(delta))
| end if
| board → update on wall
```

**end event**

**data** board : Board

**data** controls : Sprite Set

**data** darker : Sprite

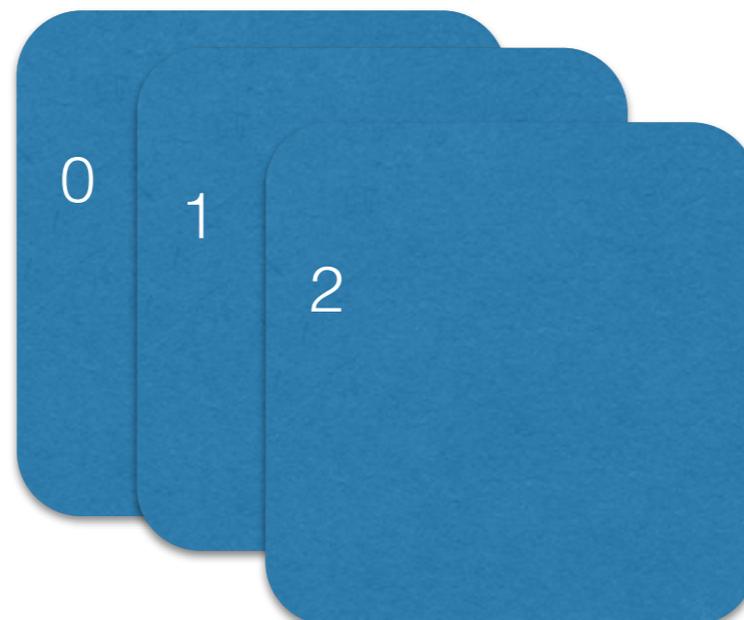
**data** lighter : Sprite

**data** rectangle : Sprite

# Pushing and popping pages (/qsht)

```
action main ()  
  |  $\square$ add := create  $\rightarrow$  Collection of  $\rightarrow$  Page Button  
  |  $\square$ back := create  $\rightarrow$  Collection of  $\rightarrow$  Page Button  
  |  $\triangleright$ fillPage  
end action
```

```
action fillPage ()  
  | wall  $\rightarrow$  clear  
  |  $\square$ page  $\rightarrow$  post to wall  
  |  $\square$ add  $\rightarrow$  add(wall  $\rightarrow$  add button("add", "push"))  
  |  $\square$ back  $\rightarrow$  add(wall  $\rightarrow$  add button("back", "pop"))  
end action
```



```
event tap wall Page Button (  
  | item : Page Button)  
do  
  | if  $\square$ add  $\rightarrow$  contains(item) then  
    | wall  $\rightarrow$  push new page  
    |  $\square$ page :=  $\square$ page + 1  
    |  $\triangleright$ fillPage  
  | else do nothing end if  
  | if  $\square$ back  $\rightarrow$  contains(item) then  
    | wall  $\rightarrow$  pop page  
    |  $\square$ page :=  $\square$ page - 1  
  | else do nothing end if  
end event  
  
data add : Collection of Page Button  
  
data back : Collection of Page Button  
  
data page : Number
```

# TouchDevelop's 'reflection'

```
action main ()
```

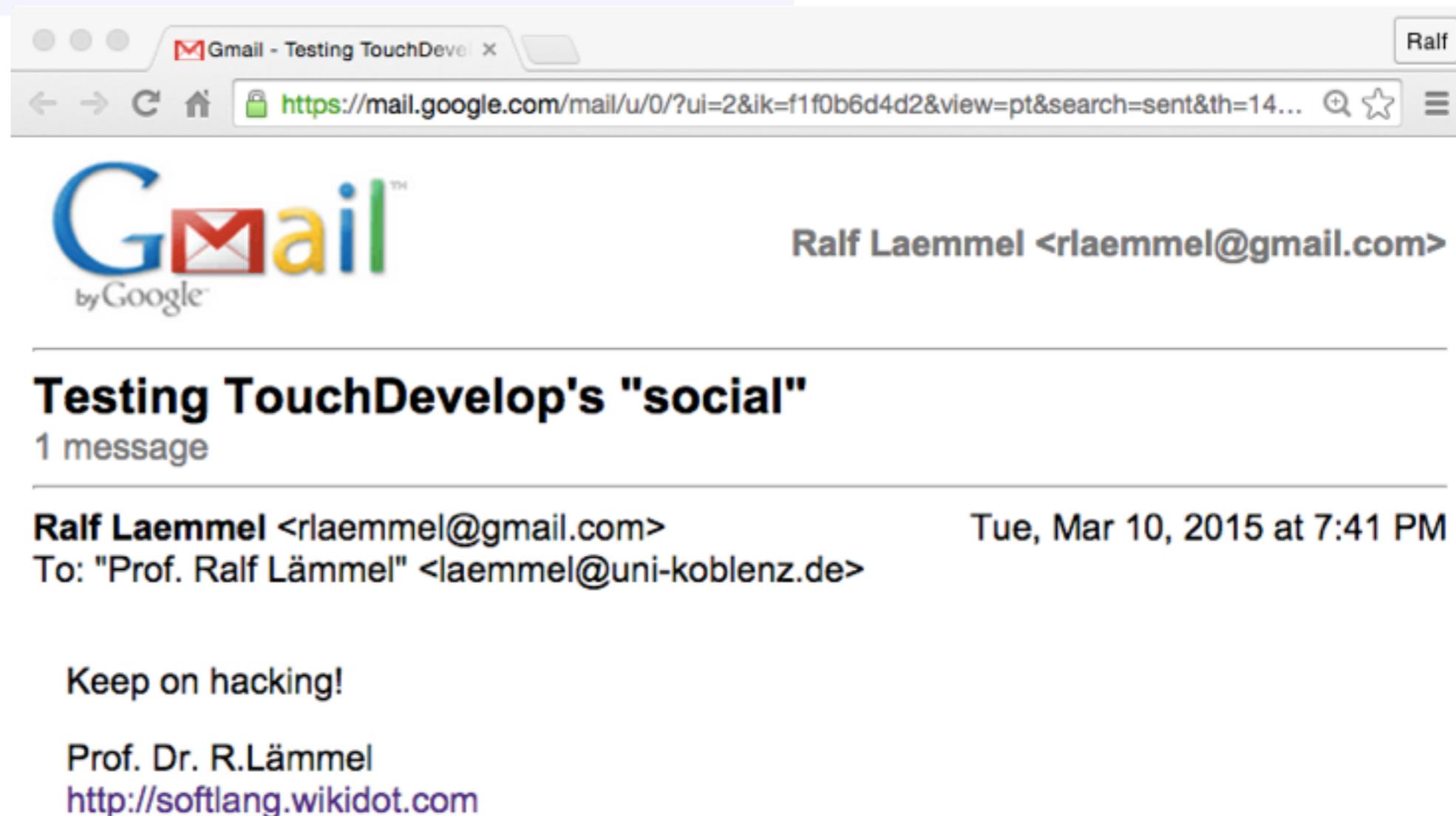
```
| wall → button icon names → post to wall
```

```
end action
```

[add, back, cancel, check, close, delete, download, edit, favs.addto, favs, feature.camera, feature.email, feature.search, feature.settings, feature.video, folder, minus, new, next, questionmark, refresh, save, share, stop, sync, transport.ff, transport.pause, transport.play, transport.rew, upload]

# Sending an email

```
action main ()
  var to := "laemmel@uni-koblenz.de"
  var subject := "Testing TouchDevelop's \"social..."
  var body := "Keep on hacking!"
  social → send email(to, subject, body)
end action
```



# Browse the Web (/emvc)

**action** main ()

| web→ link url("Go to TouchDevelop", "https://www.touchdevelop.com/")→ post to wall

| web→ link url("Go to YouTube", "https://www.youtube.com/watch?v=rn-wj4pRplE")→ post to wall

**end** action

# Language characteristics

- Statement orientation („:=“, if, while, for, ...)
- Static type checking and strong typing
- Value and reference types
- Garbage-collected heap
- No class inheritance or method overloading
- Event-handling style
- Single wall- and multi page-based user interfaces
- Mutable and immutable collections
- ...

THE EXPERT'S VOICE® IN MOBILE TECHNOLOGIES

Microsoft Research

THIRD EDITION

# TouchDevelop

Programming on the Go

Nigel Horspool and Nikolai Tillmann

Apress  
open

Available  
for free!

[https://  
az31353.vo.msecnd.net/cpd/  
btih-book.pdf](https://az31353.vo.msecnd.net/cpd/btih-book.pdf)

# TouchDevelop. Programming on the Go.

## **Chapter 3      The Wall – using the screen**

- 3.1    Output – the writing on the wall
- 3.2    Input of values from the touchscreen
- 3.3    Updating the wall's content
- 3.4    Events on the touchscreen
- 3.5    Pushing and popping pages
- 3.6    Titles and subtitles
- 3.7    Wall buttons
- 3.8    On-demand creation of output

## **Chapter 4      The Web**

- 4.1    URLs and webpages
- 4.2    Downloading and uploading files
- 4.3    Downloading structured data
- 4.4    REST guidelines and web requests

## **Chapter 5      Audio**

- 5.1    Music
- 5.2    Sounds
- 5.3    Microphone

## **Chapter 6      Camera, Graphics and Video**

- 6.1    Camera
- 6.2    Working with pictures
- 6.3    Static graphics drawing and display
- 6.4    Playing videos from the internet

## **Chapter 7      Sensors**

- 7.1    The sensors
- 7.2    Sensor-driven events
- 7.3    Accelerometer
- 7.4    Compass
- 7.5    Gyroscope

# TouchDevelop. Programming on the Go.

## **Chapter 8      Interactions**

- 8.1 Social messages
- 8.2 Locations, places, maps
- 8.3 Emails
- 8.4 Phone Calls
- 8.5 2D barcodes
- 8.6 SMS messages (WP8 only)
- 8.7 Calendar and appointments (WP8 only)
- 8.8 Contacts (WP8 and Android only)

## **Chapter 9      Game Board**

- 9.1 Introduction
- 9.2 The Board datatype
- 9.3 The Sprite datatype
- 9.4 The Sprite Collection datatype
- 9.5 Touching and board events
- 9.6 Debugging games

## **Chapter 10     UI with Boxes and Pages**

- 10.1 Page Overview
- 10.2 Box Overview
- 10.3 Examples of Boxes and Pages
- 10.4 Working with Pages
- 10.5 Live Editing of the User Interface
- 10.6 API Support for Boxes and Pages

## **Chapter 11     Authenticating Web Services**

- 11.1 Registering your app
- 11.2 Authenticating
- 11.3 Libraries
- 11.4 Advanced topics

Some tables from TouchDevelop  
book serving as indicators of  
concepts and features

# Special script symbols

Symbol	Unicode Value	Description
→	U+2192	Select a method or field belonging to the value provided on the left
▷	U+25B7	Call the action named on the right and defined in the current script
☐	U+25F3	Access a global persistent variable defined in the data section of the script
♻️	U+267B	Call a function defined in another script, which has been published as a library
🗄️	U+2339	Access a datatype or item declared in the record section of the script
🌸	U+273F	Access a value in the art section of the script

# Value types

Value Type	Description	Covered in Chapter
Number	An integer or floating-point number	2
Boolean	The type whose constants are true and false	2
String	A sequence of zero or more Unicode characters	2
Color	Used for colors displayed on the screen. Values are compatible with 4 byte ARGB (alpha, red, green, blue) color representations. Many standard colors are provided as constants of the Color datatype.	6
DateTime	Holds any date from 0001 CE (Common Era) to 9999 CE, combined with a time of day. The time of day is recorded with 100 nanosecond accuracy.	8
Location	Holds a combination of latitude, longitude and altitude values plus a course direction and a speed in two-dimensional space.	8
Motion	A combination of sensor readings which describe motion of the phone in 3D space plus a time-stamp which specifies when the readings were taken. The motion information includes speed, acceleration and angular velocity.	7
Vector3	A triple of three numbers used to hold a velocity or acceleration in the three spatial dimensions or an angular velocity about the three axes in 3D space.	7

# Reference types by the API

Reference Type	Description	Storage
Appointment	A calendar appointment	Heap
Board	A 2D canvas on which sprites can be drawn and moved	Heap
Camera	The front or back camera	External
Contact	Contact information and details about a person or company	External
Form Builder	Used to create HTML form data	Heap
Json Builder	A JSON data structure builder	Heap
Json Object	A JSON data structure (obtained from a website)	Heap
Link	A link to a video, image, e-mail or phone number	Heap
Location	A geographic location	Heap
Map	A BING map	Heap

Matrix	a 2-D matrix of numbers	Heap
Media Link	A media file on the home network	External
Message	A posting on a message board	Heap
OAuth Response	OAuth 2.0 access token or error	Heap
Page	A page on the wall	Heap
Page Button	A button on the wall which can be tapped	Heap
Picture	A rectangular picture containing graphics or a photograph	External
Picture Album	A named album of pictures	External
Place	A named location	Heap
Playlist	A song playlist	Heap
Song	A song	External
Song Album	A song album	External
Sound	A sound clip	Heap
Sprite	A graphical object which can be displayed on a Board instance	Heap
TextBox	A box used to display text on the screen	Heap
Web Request	A HTTP web request	Heap
Web Response	A HTTP web response	Heap
Xml Object	A XML element or collection of elements	Heap

# Regular collection types

Collection Type	Element Type	Mutable?
Appointment Collection	Appointment	No
Contact Collection	Contact	No
Link Collection	Link	Yes
Location Collection	Location	Yes
Media Link Collection	Media Link	No
Message Collection	Message	Yes
Number Collection	Number	Yes
Page Collection	Page	No
Picture Albums	Picture Album	No
Pictures	Picture	No
Place Collection	Place	Yes
Playlists	Playlist	No
Song Albums	Song Album	No
Songs	Song	No
String Collection	String	Yes

# Operators on numbers etc.

Operator	Operand Types	Result Type	Description
+	Number	Number	prefix unary plus
-	Number	Number	prefix unary minus
+	Number Number	Number	addition
-	Number Number	Number	subtraction
*	Number Number	Number	multiplication
/	Number Number	Number	division
<	Number Number	Boolean	less-than comparison
≤	Number Number	Boolean	less-than-or-equal comparison
>	Number Number	Boolean	greater-than comparison
≥	Number Number	Boolean	greater-than-or-equal comparison
=	Number Number	Boolean	equals comparison
≠	Number Number	Boolean	not equals comparison
not	Boolean	Boolean	logical negation
and	Boolean Boolean	Boolean	logical and
or	Boolean Boolean	Boolean	logical or
	any any	String	string concatenation

# Events (at device level specifically)

Event	Description of when invoked
gameloop	Invoked every 50 milliseconds. Intended for updating a game display, but usable for other purposes.
shake	Device is shaken
phone face up	Device is turned to be flat and face up
phone face down	Device is turned to be flat and face down
phone portrait	Device is turned upright
phone landscape left	Device is placed on its left side
phone landscape right	Device is placed on its right side
empty space on wall	The user has scrolled to the bottom of the wall and/or space to display new output has become available
page navigated from	The page displayed on the screen has been popped
active song changed	The song being played by the phone has changed
player state changed	The media player starts playing
camera button pressed	Camera button on phone is pressed
camera button half pressed	Camera button on phone is halfway pressed
camera button released	Camera button on phone is released
tap wall <i>XXX</i> (item: <i>XXX</i> )	A value of type <i>XXX</i> displayed on the wall is tapped; this value is received as the input parameter. There is a different tap wall event for various types, represented by <i>XXX</i> .

# Gameboard events

Event	Description
tap board(x: Number, y: Number)	The gameboard has been tapped at coordinates x, y.
swipe board(x: Number, y: Number, delta x: Number, delta y: Number)	The gameboard has been swiped, starting at coordinates x, y and continuing along the vector delta x, delta y.
tap sprite in sprite set(sprite: Sprite, index in set: Number, x: Number, y: Number)	A sprite in the sprite set has been tapped; the first two parameters specify which sprite and the last two give the x, y coordinates of the sprite.
swipe sprite in sprite set(sprite: Sprite, index in set: Number, x: Number, y: Number, delta x: Number, delta y: Number)	A sprite in the sprite set has been swiped; the first two parameters specify which sprite, the next two give the x, y coordinates of the sprite, and the last two give the vector describing the swipe motion.
drag sprite in sprite set(sprite: Sprite, index in set: Number, x: Number, y: Number, delta x: Number, delta y: Number)	A sprite in the sprite set has been dragged; the first two parameters specify which sprite, the next two give the x, y coordinates of the sprite, and the last two give the vector describing the dragging motion.

# Display of *media* values

Datatype	What is displayed
Picture	The picture, resized if necessary to fit the screen.
Board	The board (note that the board can be changed and redisplayed dynamically).
Song	A play button plus whichever of these items is available: duration, artist, name of album from which the song was obtained, the album cover, track number.
Sound	The text "A sound..." and a button to play the sound.
Picture Album	A sequence of all pictures in the album.
Song Album	A play button plus whichever of these items if available: total duration, artist, name of album, the album cover, number of tracks.

# Display of *social* values

Datatype	What is displayed
Appointment	The date, time and details of the appointment.
Contact	The name of the contact plus buttons which can be tapped to initiate a phone call or send a SMS message or send an email to this contact .
Link	The name associated with the link plus a button to initiate a phone call, send a SMS message or send an email, depending on the kind of link.
Location	A scrollable Bing map which shows the location.
Message	The name of the sender, the time when the message was sent plus the contents of the message.
Place	The name associated with the place plus a thumbnail map showing the location of the place.

# TouchDevelop resources

- <https://www.touchdevelop.com/>
- <https://www.touchdevelop.com/app/>
- <https://www.touchdevelop.com/docs/faq>
- <https://az31353.vo.msecnd.net/cpd/btih-book.pdf>
- <https://az31353.vo.msecnd.net/cpd/qkxj-tutorialwalkthrough.pdf>
- <http://www.drdoobs.com/embedded-systems/building-android-apps-with-touchdevelop/240165507>
- ...