PROGRAMMIERUNG - PROJEKTARBEIT

2025, Nikolaus Albrecht

Skript: http://www.sigmadelta.at/

INHALT:

1_Organisation, Ablauf und was Sie erwartet

Termine

Fr. 21.11.2025 | 14:30 bis 17:45 | virtueller Raum | AN & SLJ

Sa. 22.11.2025 | 08:45 bis 16:45 | D.2.002 | AN & SLJ

Fr. 12.12.2025 | 14:30 bis 17:00 | virtueller Raum | AN, SLJ, FK, +

Informationen bezüglich ...

Präsentationstermin (12.12.25):

- Zwei Breakout-Rooms 15 Minuten Präsentation pro Person (inklusive Feedback und Wechsel)

Abgabe:

- Projektarbeit über TSN-Moodle

Kontaktperson:

- Kathrin Jarosik (kathrin.jarosik@ph-tirol.ac.at)

Details:

Für Freitag:

Zweiteilung (Inputphase und Rückfragemöglichkeiten für Projektarbeit / Präsentation)

Für Samstag:

Erster Runde: von 8:45 bis 12:00

Mittagessen: von 12:00 bis 13:30

Zweite Runde: von 13:30 bis 16:45

Projektpräsentation-Probelauf mit Feedback und Rückfragemöglichkeit

2_LEHRPLANBEZUG

Bundesministerium
Bildung, Wissenschaft
und Forschung

DIGITALE GRUNDBILDUNG (Sekundarstufe I)

Bildungs- und Lehraufgabe (1. bis 4. Klasse):

Die Digitale Grundbildung bietet vielfache Möglichkeiten zum fächerverbindenden Unterricht, unter der besonderen Berücksichtigung der Schnittstellen von der digitalen zur analogen Welt. Lehrende sind angehalten, unmittelbare Konsequenzen der Digitalisierung für Wissen und Bildung zu berücksichtigen sowie aktuelle Themen und Entwicklungen kritisch und reflektiert aufzugreifen.

- Medienbildung umfasst die Beschäftigung mit der Entstehung, Entwicklung und Zukunft digitaler Medienkonstellationen. Reflexion und Kritik betreffen beispielsweise medienbiografische Entwicklungen bzw. Bedingungen der Mediensozialisation sowie digitaler Inklusions- und Exklusionsdynamiken. Medienbildung geht vom Zusammenspiel von Nutzung und Teilnahme an aktueller Medienkultur aus.
- Informatische Bildung umfasst das Analysieren, Interagieren, Modellieren, Codieren und Testen im Umgang mit Informatiksystemen, Software, Automatisierung, Daten und Vernetzung. Die Entwicklung informatischer und medientechnischer Kompetenzen orientiert sich besonders an didaktischen Prinzipien der sogenannten 21st Century Skills, der 4 Ks (kritisches Denken, Kreativität, Kommunikation und Kollaboration) und des Computational Thinking (problemorientiertes informatisches Denken).
- Gestaltungskompetenz geht von einem Zusammenspiel von informatischer Bildung und Medienbildung aus und bietet vielfältige analytische, produktive und kreative Zugänge zu funktionalen Medieneinsätzen und ästhetischen Medienformaten in globalisierten digitalen Kulturen.

Kompetenzmodell und Kompetenzbereiche (1. bis 4. Klasse):

- **Orientierung:** gesellschaftliche Aspekte von Medienwandel und Digitalisierung analysieren und reflektieren
- **Information:** mit Daten, Informationen und Informationssystemen verantwortungsvoll umgehen
- **Kommunikation:** Kommunizieren und Kooperieren unter Nutzung informatischer, medialer Systeme
- **Produktion:** Inhalte digital erstellen und veröffentlichen, Algorithmen entwerfen und Programmieren: Zerlegen von Problemen, Muster erkennen, Verallgemeinern/Abstrahieren und Algorithmen entwerfen
- **Handeln:** Angebote und Handlungsmöglichkeiten in einer von Digitalisierung geprägten Welt einschätzen und verantwortungsvoll nutzen

Kompetenzbeschreibungen und Anwendungsbereiche, Lehrstoff (1. bis 4. Klasse):

1. Klasse:

Kompetenzbereich Orientierung: gesellschaftliche Aspekte von Medienwandel und Digitalisierung analysieren und reflektieren

Die Schülerinnen und Schüler können

- (T) das Prinzip der Eingabe, Verarbeitung und Ausgabe exemplarisch an den Bestandteilen und der Funktionsweise eines digitalen Endgeräts beschreiben.

Kompetenzbereich Produktion: Inhalte digital erstellen und veröffentlichen, Algorithmen entwerfen und Programmieren

Die Schülerinnen und Schüler können

- (T) eindeutige Handlungsanleitungen (Algorithmen) nachvollziehen, ausführen sowie selbstständig formulieren.
- (G) verschiedene Darstellungsformen von Inhalten und die Wirkung auf sich und andere beschreiben.[7]
- (I) mit Daten einfache Berechnungen durchführen sowie in verschiedenen (visuellen) Formaten sammeln und präsentieren.
- (I) einzeln und gemeinsam Texte und Präsentationen (unter Einbeziehung von Bildern, Grafiken und anderen Objekten) strukturieren und formatieren.[10]

Anwendungsbereiche

- Sequenzen und einfache Schleifen
- Planung, Gestaltung und Auswertung von Umfragen

2. Klasse:

Kompetenzbereich Information: mit Daten, Informationen und Informationssystemen verantwortungsvoll umgehen

Die Schülerinnen und Schüler können

- (T) Daten erfassen, filtern, sortieren, interpretieren und darstellen.
- (T) beschreiben, wie über das Internet Informationen bereitgestellt und abgerufen sowie Daten übertragen werden.
- (I) Lizenzmodelle, insb. offene (Creative Commons, Open Educational Resources, Open Source), benennen, erklären und anwenden.[13]

Anwendungsbereiche

- Organisation von Daten
- (Visuelle) Darstellung von Daten
- Beschreibung von Daten hinsichtlich ihrer Formate, Größe und binären Struktur

Kompetenzbereich Produktion: Inhalte digital erstellen und veröffentlichen, Algorithmen entwerfen und Programmieren

Die Schülerinnen und Schüler können

- (T) darstellen, wie Programme Daten speichern und verarbeiten, indem sie Zahlen oder andere Symbole zur Darstellung von Informationen verwenden.
- (T) unter Nutzung einer geeigneten Entwicklungsumgebung einfache Programme erstellen, diese testen und debuggen (Fehler erkennen und beheben).
- (G) die Rechte am geistigen Eigentum beachten und bei der Erstellung oder beim Remixen von Programmen die entsprechenden Urheberrechte angeben.[13]
- (I) visuelle/audiovisuelle/auditive Inhalte erzeugen, adaptieren und analysieren. Sie können Möglichkeiten der Veröffentlichung benennen.

3. Klasse:

Kompetenzbereich Information: mit Daten, Informationen und Informationssystemen verantwortungsvoll umgehen

Die Schülerinnen und Schüler können

- (G) Bedingungen sowie Vor- und Nachteile von personalisierten Suchroutinen für das eigene Leben bzw. die Gesellschaft erklären.
- (I) zielgerichtet und selbstständig die Suche nach Informationen und Daten mit Hilfe geeigneter Strategien und Methoden planen und durchführen, geeignete Quellen nutzen und gefundene Informationen vergleichend hinterfragen.
- (I) Muster in Datendarstellungen wie Diagrammen oder Grafiken erkennen und beschreiben, um Vorhersagen zu treffen.
- (I) Datenmaterial nutzen, um Ursache-Wirkung-Beziehungen aufzuzeigen oder vorzuschlagen, Ergebnisse vorherzusagen oder eine Idee zu vermitteln.

Anwendungsbereiche

- Manipulative und monoperspektivische Darstellungen von Informationen in populären Medienkulturen[7]

Kompetenzbereich Produktion: Inhalte digital erstellen und veröffentlichen, Algorithmen entwerfen und Programmieren

Die Schülerinnen und Schüler können

- (T) an Beispielen Elemente des Computational Thinkings nachvollziehen und diese zur Lösung von Problemen einsetzen. Sie wissen, wie sie Lösungswege in Programmiersprache umsetzen können.
- (G) verschiedene populäre Medienkulturen benennen sowie Möglichkeiten verschiedener Darstellungsformen von Inhalten erproben.
- (I) ihre eigenen medialen Produktionen auf Barrierefreiheit überprüfen und ggf. Barrieren beseitigen.[8]
- (I) Einstellungen in Softwareapplikationen den persönlichen Bedürfnissen entsprechend anpassen.

Anwendungsbereiche

- Gezielte bzw. manipulative Darstellungen, zB in Diagrammen, durch Bildausschnitte oder Vertonung[7]
- Konfigurationsmöglichkeiten von Betriebssystemen und Kommunikationssystemen, um sie barrierefrei zugänglich machen[8]

4. Klasse:

Kompetenzbereich Produktion: Inhalte digital erstellen und veröffentlichen, Algorithmen entwerfen und Programmieren

Die Schülerinnen und Schüler können

- (T) Programme entwerfen und iterativ entwickeln, die Kontrollstrukturen kombinieren, einschließlich verschachtelter Schleifen und zusammengesetzter Konditionale.
- (G) Möglichkeiten verschiedener Darstellungsformen von Inhalten erproben und deren Einfluss auf die Wahrnehmung des Inhalts hinterfragen.
- mit bereitgestellten Medien und Software-Applikationen zielgerichtet und kreativ gestaltend kooperieren.
- einfache Programme oder Webanwendungen mit geeigneten Werkzeugen erstellen, um ein bestimmtes Problem zu lösen oder eine bestimmte Aufgabe zu erfüllen.

Anwendungsbereiche

- Dokumentation von Programmen[10]
- Projektplanung inklusive Aufgabenverteilung und Zeitplan[1, 2]
- (Graphische) Notationen, Pseudocode
- Ästhetische und technische Kompetenzen von Medienkulturen in Projekten

3_Storytelling & Scratch & Competition

Storytelling-Competition

Subject: Invitation: 1st European School Map Storytelling Competition

Dear Educator,

Following your training in the framework of **GEO-Academy**, we are happy to invite you to the **1st European School Map Storytelling Competition**!

In the European Map Storytelling Competition, your task is to identify a story that **connects** your **school** or wider community to at least **one of the 17 Sustainable Development Goals (SDGs)**, and tell that story using maps.

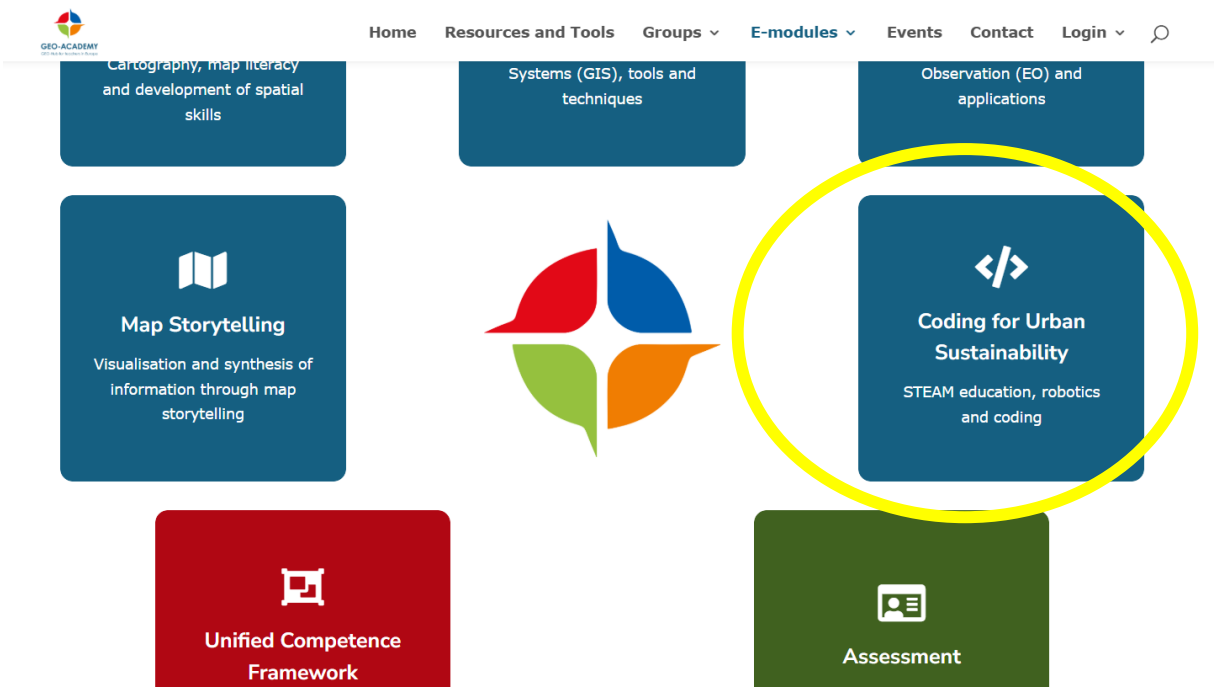
Learn more about the competition here:
https://geoacademy.eu/mapstorytelling-competition/

We look forward to your participation!

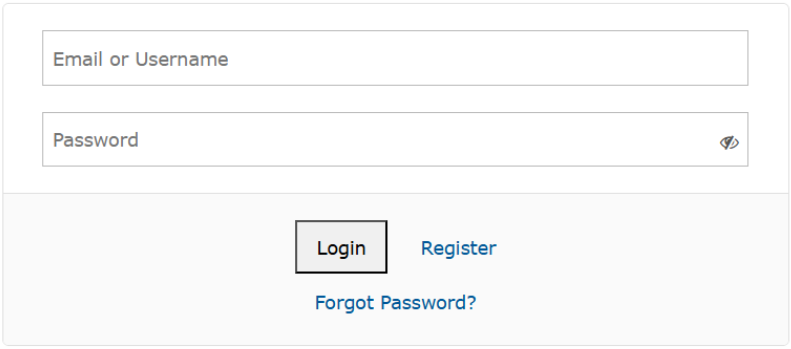
Official competition page:GeoVoices: Map Storytelling for Global Sustainability Goals – Geo Academy

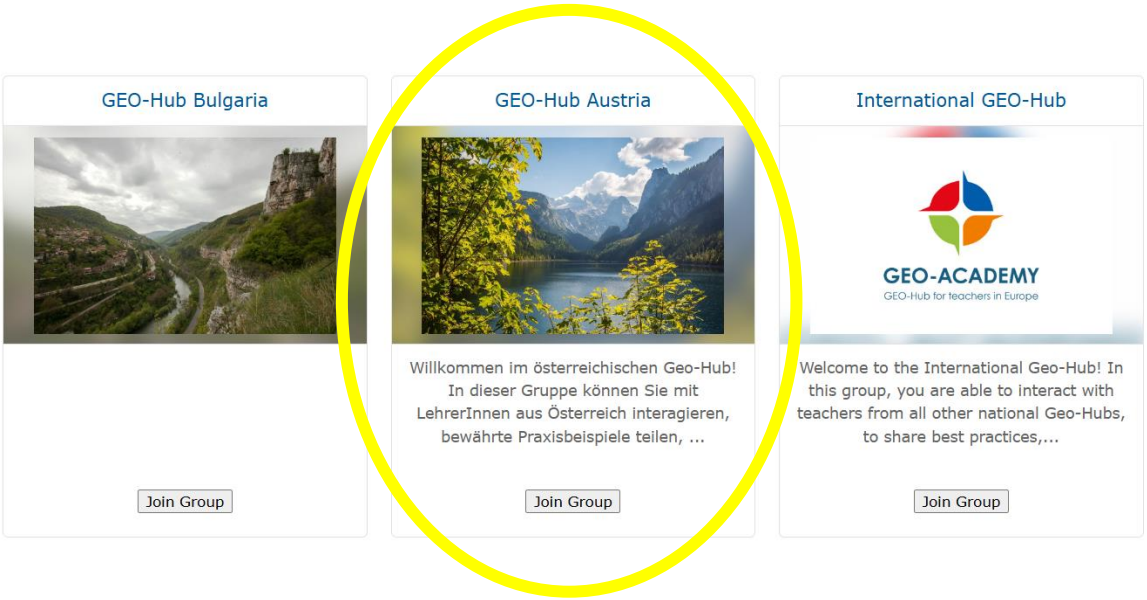
Kontaktadresse: n.albrecht@ph-tirol.ac.at

https://portal.geoacademy.eu/e-modules/



Login


Email or Username
Password
Login
Register
Forgot Password?


GEO-Hub Bulgaria
Join Group
GEO-Hub Austria
Willkommen im österreichischen Geo-Hub! In dieser Gruppe können Sie mit LehrerInnen aus Österreich interagieren, bewährte Praxisbeispiele teilen, ...
Join Group
International GEO-Hub
GEO-ACADEMY
GEO-Hub for teachers in Europe
Welcome to the International Geo-Hub! In this group, you are able to interact with teachers from all other national Geo-Hubs, to share best practices,...
Join Group

Geo-concept: STEAM education, robotics, and coding

Search courses...

Geo-concept: STEAM education robotics and coding

Robotics And Coding Team

STEAM education, robotics, and coding – Part 4

Free

Geo-concept: STEAM education robotics and coding

Robotics And Coding Team

STEAM education, robotics, and coding – Part 3

Free

Geo-concept: STEAM education robotics and coding

Robotics And Coding Team

STEAM education, robotics, and coding – Part 2

Free

Geo-concept: STEAM education robotics and coding

Robotics And Coding Team

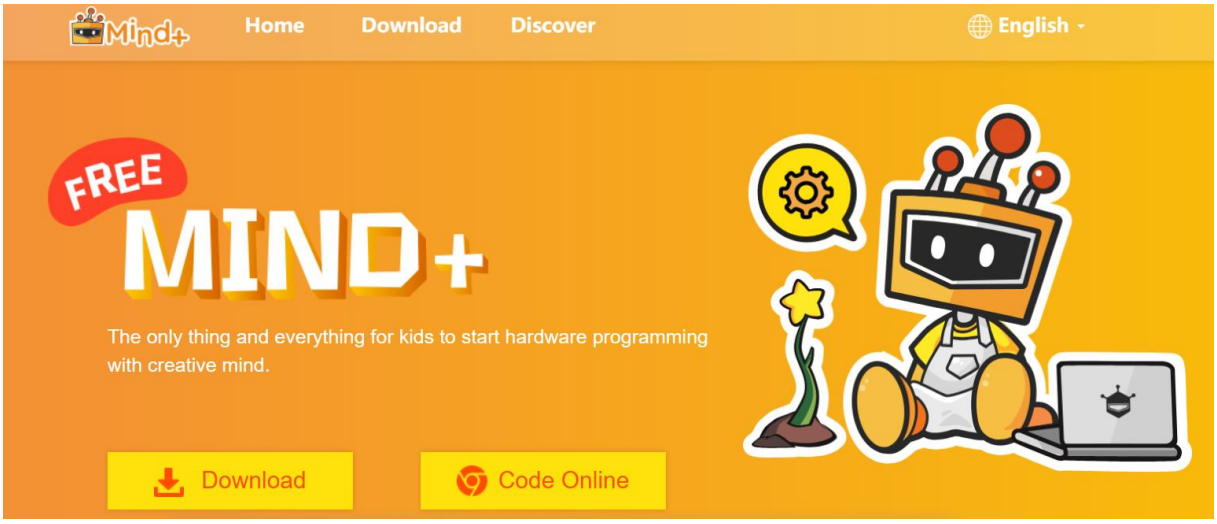
STEAM education, robotics, and coding – Part 1

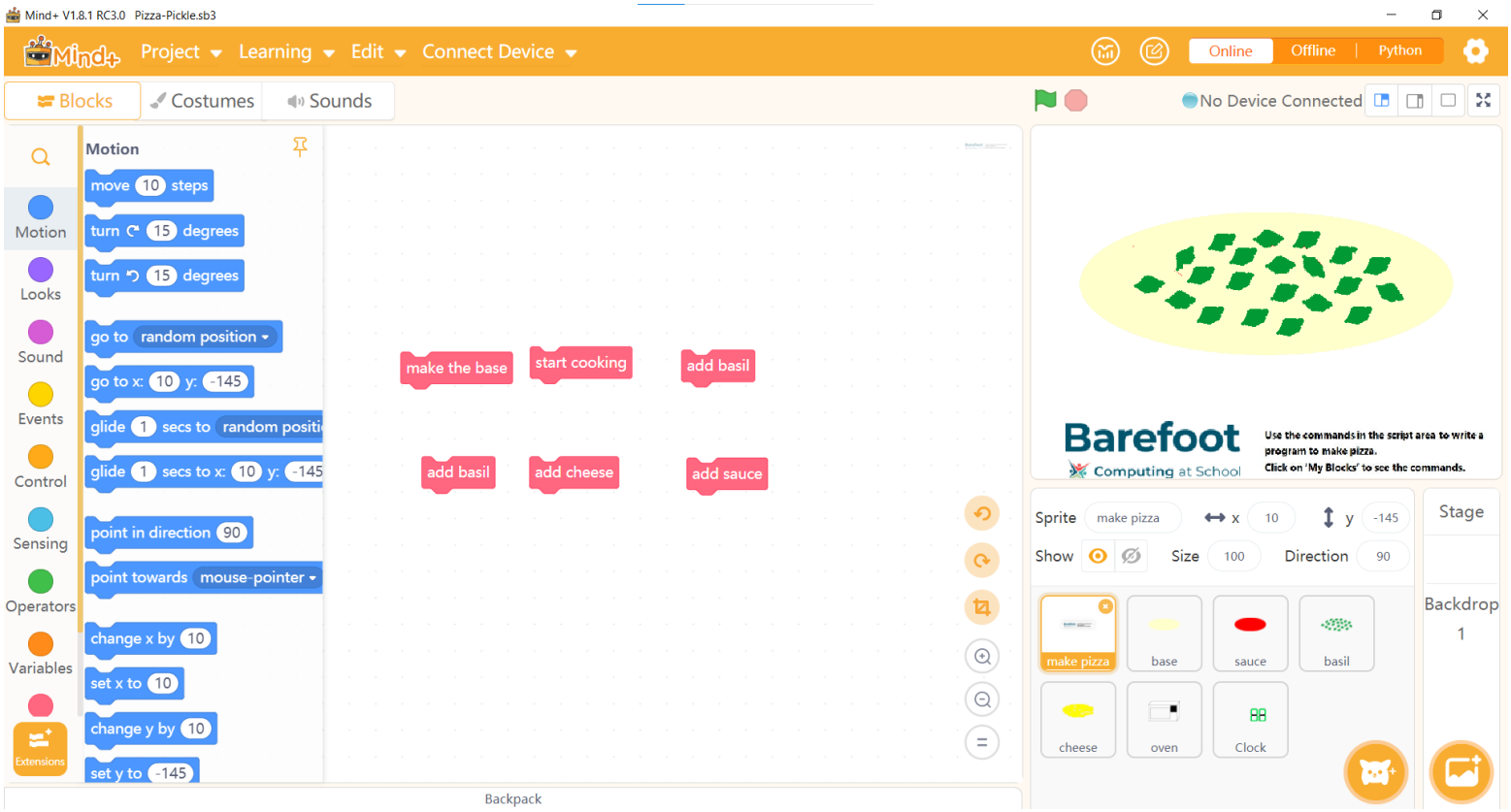
Free

4_MIND+

Mind+

Official Website : http://mindplus.cc




Mind+ V1.8.1 RC3.0 Pizza-Pickle.sb3
Project
Learning
Edit
Connect Device
Online
Offline
Python
Blocks
Costumes
Sounds
No Device Connected
Motion
Looks
Sound
Events
Control
Sensing
Operators
Variables
Extensions
move 10 steps
turn 15 degrees
turn 15 degrees
go to random position
go to x: 10 y: -145
glide 1 secs to random positi
glide 1 secs to x: 10 y: -145
point in direction 90
point towards mouse-pointer
change x by 10
set x to 10
change y by 10
set y to -145
make the base
start cooking
add basil
add basil
add cheese
add sauce
Barefoot
Computing at School
Use the commands in the script area to write a program to make pizza.
Click on 'My Blocks' to see the commands.
Sprite
make pizza
x
10
y
-145
Show
Size
100
Direction
90
Stage
Backdrop
1
make pizza
base
sauce
basil
cheese
oven
Clock
Backpack

5_Debugging Activity (Barefoot computing)

https://www.barefootcomputing.org/

PIZZA PICKLE SCRATCH DEBUGGING

Age: 7 - 9 years (although this can be adapted for other years)

Concepts / Approaches: Abstraction, Programming, Persevering, Logic, Debugging

Curriculum Links: D&T

Pupils are given programs that do not do as expected and are asked to fix them. Pupils use logical reasoning to predict what will happen, and develop their debugging skills.

Open resource

https://www.barefootcomputing.org/resources/pizza-pickle-scratch-debugging

Pizza Pickle Scratch Debugging

An introduction to debugging Scratch programs

Age: 7 - 9 years (although this can be adapted for other years)

Type: Scratch

Curriculum Links to: D&T

Computer Science Concepts: Programming

Computational Thinking Concepts & Approaches:

Abstraction

Persevering

Logic

Debugging

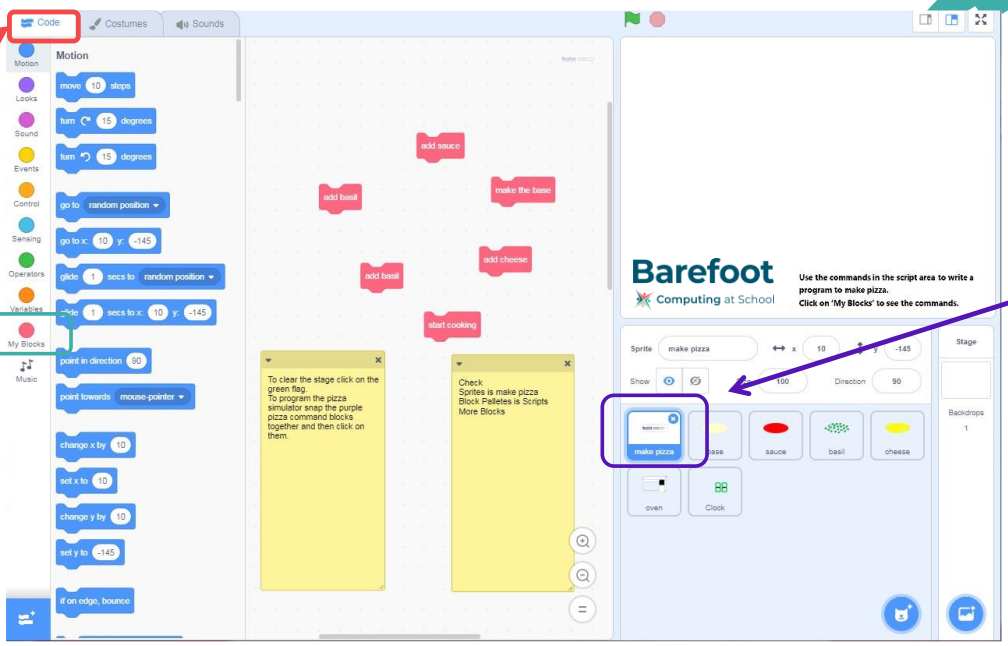
Download Resource

Add to Favourites

ONLINE ...

https://scratch.mit.edu/projects/24669802/editor/

BLOCK PALETTE | SCRIPTS AREA (WORK AREA) | STAGE AREA

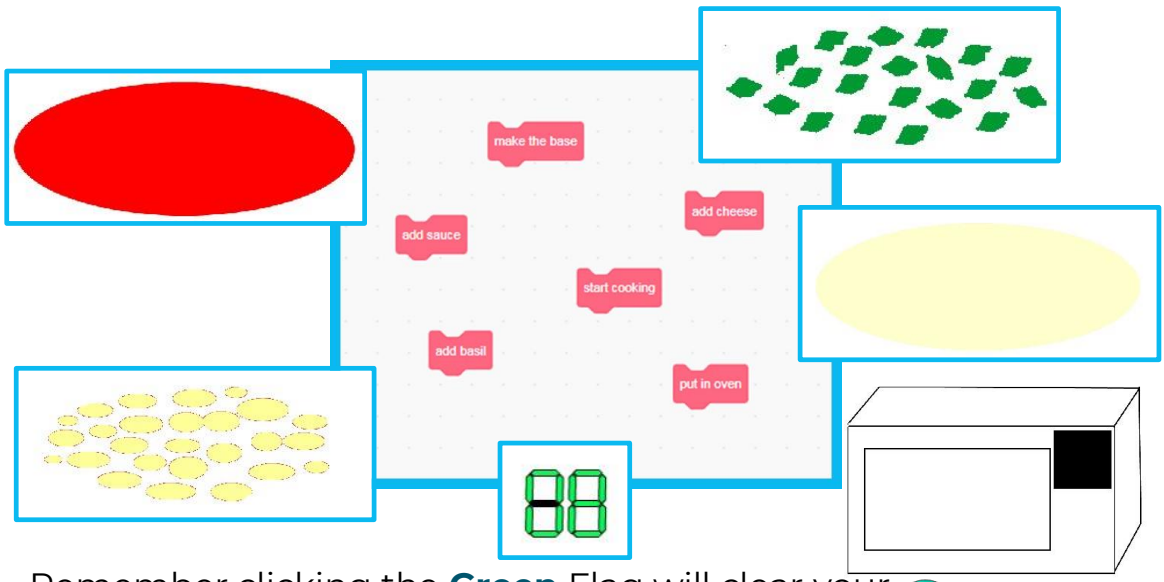


2. Click on 'Scripts'
3. Click on 'More Blocks'

1. Click on 'make pizza'

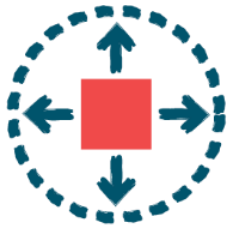
[Click here to use the Pizza Pickle Program!]

Programming Commands



Remember clicking the **Green** Flag will clear your stage

Today we are learning about...

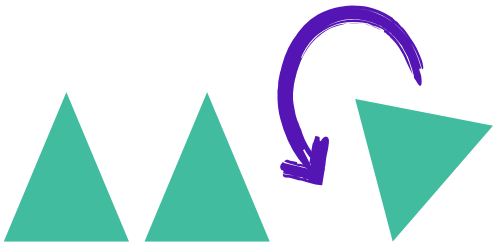


Programming

Programming

- I can debug a program
- I can say what a program will do
- I can explain what the bug was and how I fixed it

Debugging - handy hints!



1. What should it do?
2. What does it do
3. Where did it go wrong?
4. Fix it

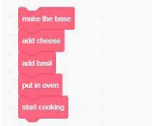
Pizza order!

1 plain pizza with no toppings:

1) What should it do?
   Make and cook a plain pizza…
2) What does it do?
3) Where does it go wrong?
4) Fix it

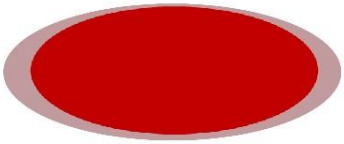
Date:_____ Program debugged by :_____

Pizza Pickle Orders

Pizza Order ***What should it do?***	Program (with bug)	**What does it do?** **Where does it go wrong?**	**Fix it!** (Your program)
Make 1 plain pizza 		It makes a base and puts it in the oven, but does not cook it.	
Make 1 pizza with sauce 			
Make 1 pizza with sauce and cheese 			
Make 1 pizza with just cheese 			

Pizza order!

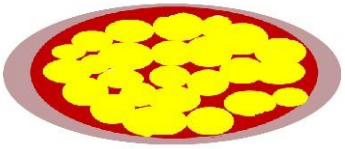
1 pizza with sauce:



1) What should it do?
   Make and cook a plain pizza with sauce...
2) What does it do?
3) Where does it go wrong?
4) Fix it

Pizza order!

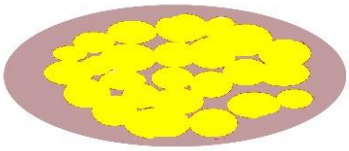
1 pizza with sauce and cheese:

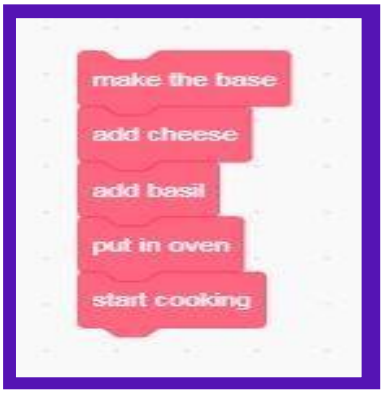


1) What should it do?
   Make and cook a pizza with sauce and cheese…
2) What does it do?
3) Where does it go wrong?
4) Fix it

Pizza order!

1 pizza with just cheese:

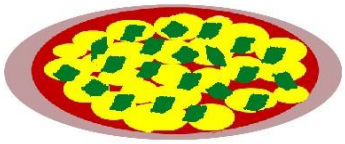


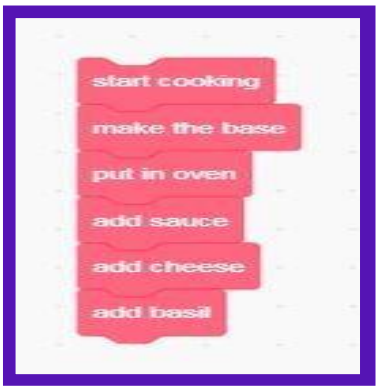


1) What should it do?
   Make and cook a pizza with just cheese…
2) What does it do?
3) Where does it go wrong?
4) Fix it

Pizza order!

1 pizza with everything:

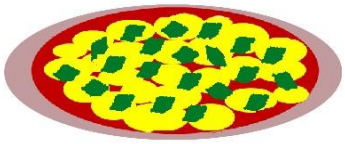




1) What should it do?
   Make and cook a pizza with everything...
2) What does it do?
3) Where does it go wrong?
4) Fix it

Pizza order!

- Make up an order for your friend
- Then program it (with a bug)
- Can you fix it?



1. What should it do?
   Make and cook a pizza with
2. What does it do?
3. Where does it go wrong?
4. Fix it

unplugged programming activities

AI Mode | All | Images | Videos | Short videos | Forums | News | More | Tools

Pdf | Free | For students | For high school | For kids | Middle School | For kindergarten

CS Unplugged
https://www.csunplugged.org

CS Unplugged

CS Unplugged is a collection of free teaching material that teaches Computer Science through **engaging games and puzzles** that use cards, string, crayons and ...

CS Unplugged | How do I teach CS Unplugged? | Topics | Printables

Code.org
https://code.org › files › Hour-of-Code-Unplugge... PDF

Hour of Code Unplugged Activity Packet

This packet includes "Everybody Dance Now" (ages 4-11), "Graph Paper **Programming**" (ages 6-13), and "Use Binary to Make Pictures" (ages 10-18) **activities**.

Kodable
https://www.kodable.com › learn › unplugged-coding-a...

Unplugged Coding Activities | The Ultimate Guide for ...

13 Nov 2020 — Unplugged Coding Activities includes **20+ projects that teach real coding concepts** while emphasizing computer science mindsets - all with minimal prep and zero ...

SARA – unplugged

The basics of coding: SARA

Sequence,
Alternative,
Repetition and
Abstraction.

SARA

Compare to mathematics is it +,-,* and / .

SARA

The basic principles of coding

SEQUENCE

ALTERNATIV

REPETITION

ABSTRACTION

SARA

To start with we do need commands:
stand , jump, spin , wave , say “Hello”, sit,

With these commands we write our first program:

SEQUENCE

stand
jump
spin
wave
“Hello”
sit

SEQUENCE

stand

jump

spin

wave

“Hello”

sit

The next one is about alternative: If you have black shoes you should jump otherwise you should turn.

If (black shoes) {jump} else {spin}

check out the colour of your shoes!

ALTERNATIV

stand

wave

“Hello”

If (black shoes) {jump} else {spin}

sit

SARA

In case of many repetitions it is better to use a new command: repeat

```
jump
spin
jump
spin
jump
spin
```

Will become:

```
repeat(3) {jump; spin}
```

REPETITION

```
stand

repeat(3) {jump ; spin}

wave

“Hello”

sit
```

Abstraction you use when you like to use the same code several times.
As for example

wave
“Hello”

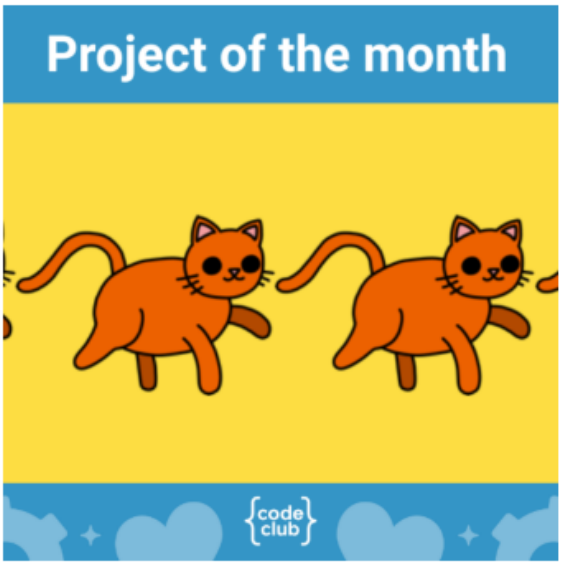
You can call this
greeting

def greeting = {wave; “Hello” }

ABSTRACTION

```
def greeting = {wave; “Hello” }

stand

greeting

repeat(3) {jump;spin}

greeting

sit
```

Discover creative projects, explore highlights from the Code Club Conference 2025, and get ready to celebrate your club's achievements with our fun new GIF!



Project of the month

The cats need your help! Coders can create a fun Scratch game where players need to draw lines to help cats reach the exit safely and avoid falling through the gaps.

Start CATS!'

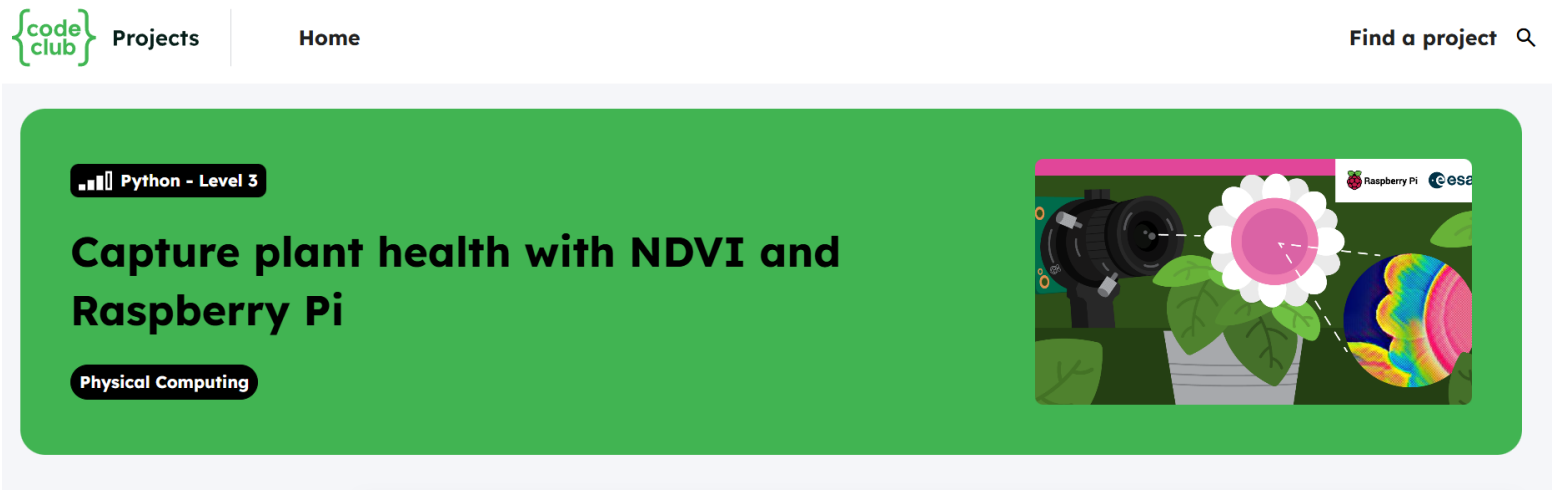
Featured resource

Check out our three handy A4 project cards, designed to be printed and used alongside a computer. Creators can use these cards to make fun projects with Scratch, the BBC micro:bit, and HTML.

See the project cards

https://projects.raspberrypi.org/en/projects/turtle-tracker

https://projects.raspberrypi.org/en/projects/astropi-ndvi



River-Crossing Activity (https://www.barefootcomputing.org/)

RIVER CROSSING ACTIVITY (SEND)

Pupils act out the traditional farmer, chicken, fox and corn problem, developing their logical reasoning skills.

Programming

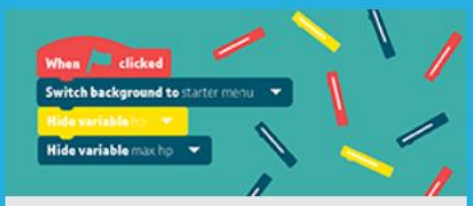
Write and test code to create software, automate tasks and solve problems using languages and tools

Scratch

These activities are based around Scratch, a visual block-based programming language, where programs are built up by dragging together command blocks.

CARBON QUEST

Pupils collect data to investigate their school's carbon emissions and program an animation in Scratch to share their findings.



SCRATCH TINKERING ACTIVITY

Pupils tinker with Scratch to find out what it does and how to create programs in it.

WORLD MAP LOGIC ACTIVITY

Pupils look at logical reasoning to predict what sequences of commands do, and then program them.

FOSSIL FORMATION ANIMATION

Pupils program animations showing fossil formation, learning to implement algorithms as code, and sequencing commands.

ACCESSIBLE ADVENTURES IN CODING

Pupils explore how an accessibility tool created in Scratch Jr can help a child with visual impairment.

SHAPES & CRYSTAL FLOWERS REPETITION

Pupils design algorithms to draw patterns made of repeating shapes before programming them in Scratch.

CLASSROOM SOUND MONITOR

Pupils create classroom sound monitors to take information from an input sensor, and alter the output of the program

Featured Resources +

Handpicked resources to aid in the classroom

Programming +

Write and test code to create software, automate tasks and solve problems using languages and tools

Micro:bit +

Programming activities using the micro:bit, a pocket sized computer with a range of inputs and outputs.

Kodu +

Learn how to implement algorithms as code through the Kodu programming environment.

Networks +

A collection of activities designed to introduce pupils to the basics of computer networks, including key parts of the system.

Data +

Activities to support the processes of gathering, organising and analysing different kinds of data.

Unplugged +

Learn about computing and coding concepts, minus the computer!

Themed Resources +

Cross curricular activities based around a range of themes including current affairs and other relevant topics.

Don't Forget! +

Short, fun activities designed to teach pupils problem-solving skills such as algorithms, decomposition and patterns

Teacher Resources +

Check out these tailor made resources

Students need to know:

1) What they are learning

2) If they've learned what they are supposed to

Learning intentions should focus on what students are LEARNING, not what they are DOING

WHAT ARE WE LEARNING TODAY?

Success Criteria

Clarity in relation to what is being taught brings about clarity in what students are expected to do to demonstrate their learning.

WARM-UP ...

Unplugged programming

The basic principles of coding

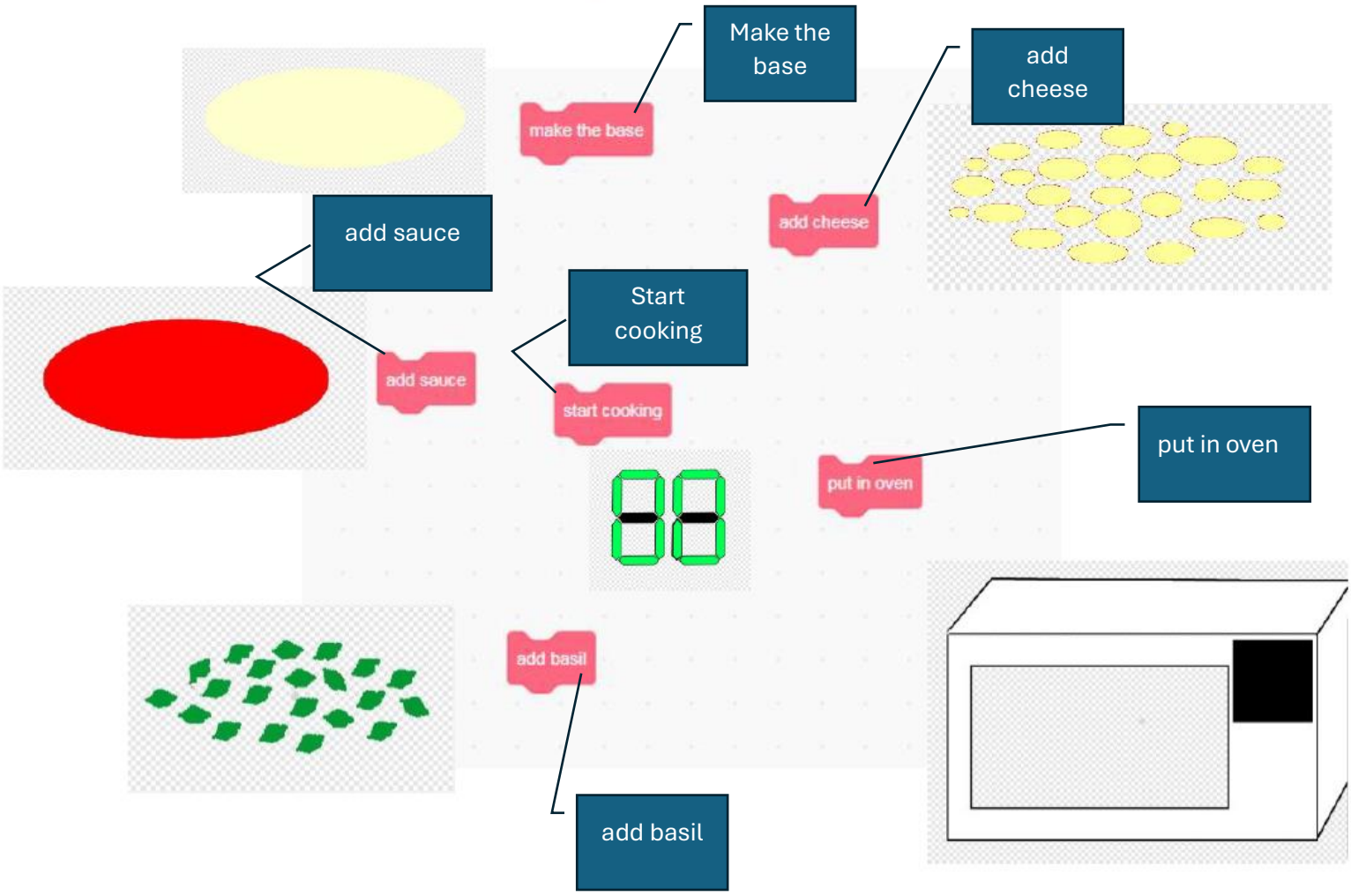
SEQUENCE

ALTERNATIV

REPETITION

ABSTRACTION

Pizza Pickle Programming Commands



Date: _____ Program debugged by : _____

Pizza Pickle Orders

Pizza Order ***What should it do?***	Program (with bug)	**What does it do? Where does it go wrong?**	**Fix it!** (Your program)
Make 1 plain pizza 		It makes a base and puts it in the oven, but does not cook it.	

make the base

put in oven

make the base

put in oven

start cooking

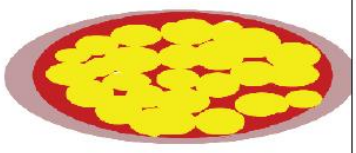
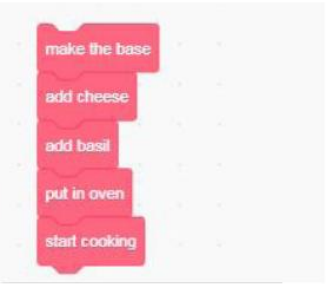
Make 1 pizza with sauce 			

make the base

add cheese

put in oven

start cooking

Make 1 pizza with sauce and cheese 			

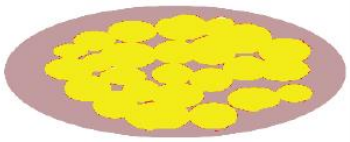
make the base

add cheese

add basil

put in oven

start cooking

Make 1 pizza with just cheese 			

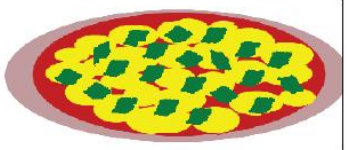
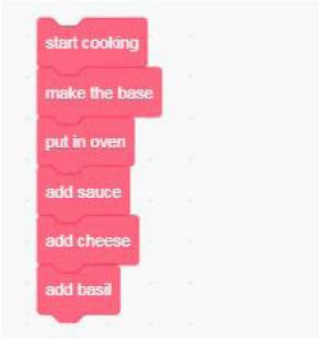
make the base

add cheese

add basil

put in oven

start cooking

Make 1 pizza with sauce, cheese and basil 			

start cooking

make the base

put in oven

add sauce

add cheese

add basil

Let's get serious about programming...

Binary search

phone book | name starts with O

log on to Facebook | account verification

Input: a sorted list of elements

Returns position

I’m thinking of a number between 1 and 100

You have to try to guess my number in the fewest tries possible.

With every guess, I tell you …

> Guess is too low
>
> Guess is too high
>
> Correct!

Suppose you start guessing like this: 1, 2, 3, 4, …

That’s a bad approach!

(“simple search”)

Worst case: 99 guesses

A better way …

Worst case: ? guesses

Success criteria ...

How about ... a list with 240.000 entries?

Simple search: ……

Binary search: …….

➔ **Running time**

➔ **Big O notation**

Linear time vs log time

Storytelling ...

... writing search algorithm for NASA

... calculating where to land.

SIMPLE SEARCH | BINARY SEARCH ?

There are only 10 seconds to figure out where to land ...

Consider the possibility of bugs in the code!

TEST RUN:

100 landing position to check

and it takes 1ms to check one element

BUT realistically, the list has 1.000.000.000 elements!

If you run the code with binary search … how long will take?

If you run the code with simple search … how long will take?

Zeit zum Entspannen ...

https://blockly.games/

https://www.roboblocky.com/

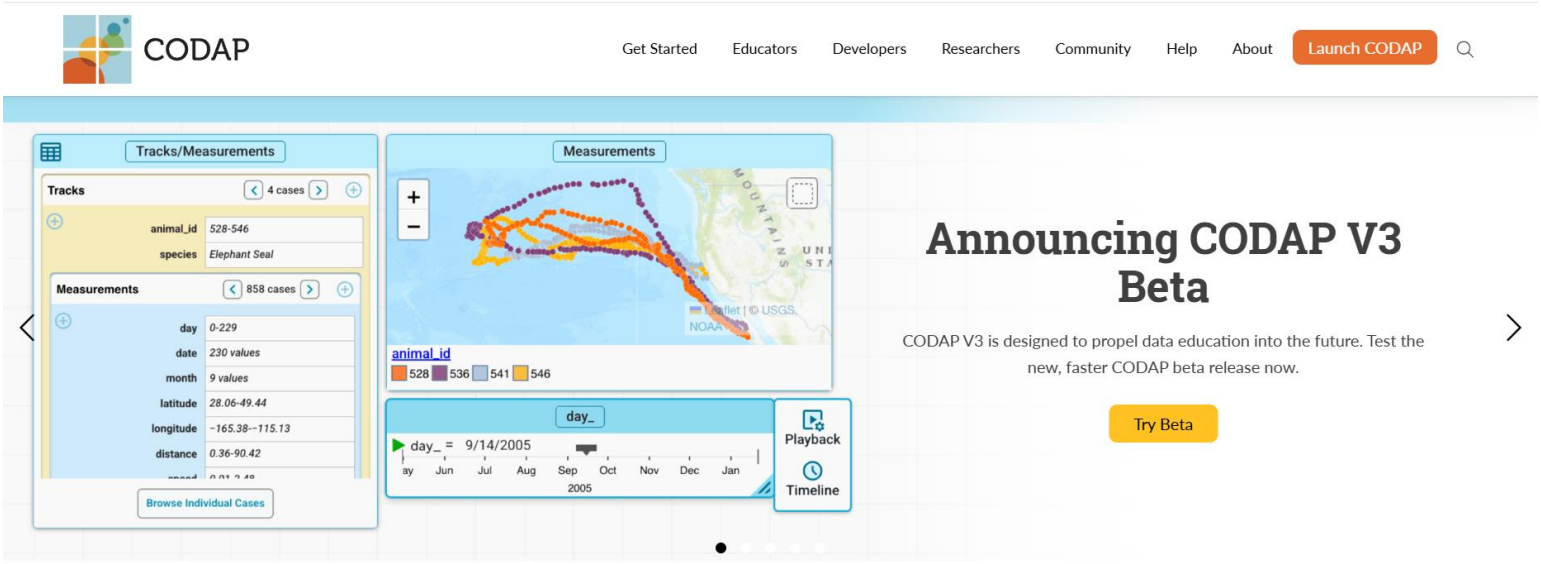
https://www.codeforlife.education/rapidrouter/

https://codecombat.com/

https://www.planet-schule.de/thema/lernspiel-programmieren-elefant-tutorials-100.html

Let's do some serious data processing...

https://codap.concord.org/



CODAP = Common ...

Online **D**ata **A**nalysis **P**latform

Of course, meanwhile you can find everything on the internet, but if – for example – the mean value (in a data set) has no meaning for you, a lot of things (out there) may be meaningless. You could be vulnerable to manipulation.

Data records made up out of (thin) air

For someone who has respiratory problems, the question of air quality can be important. A forest fire close to home could have an impact on air quality. But how can we find out?

We will analyse the air quality during and after the Bighorn Wildfire (5 June to 23 July 2020) in the nearby city of Tucson, Arizona.

To be able to do this at all, we need data.

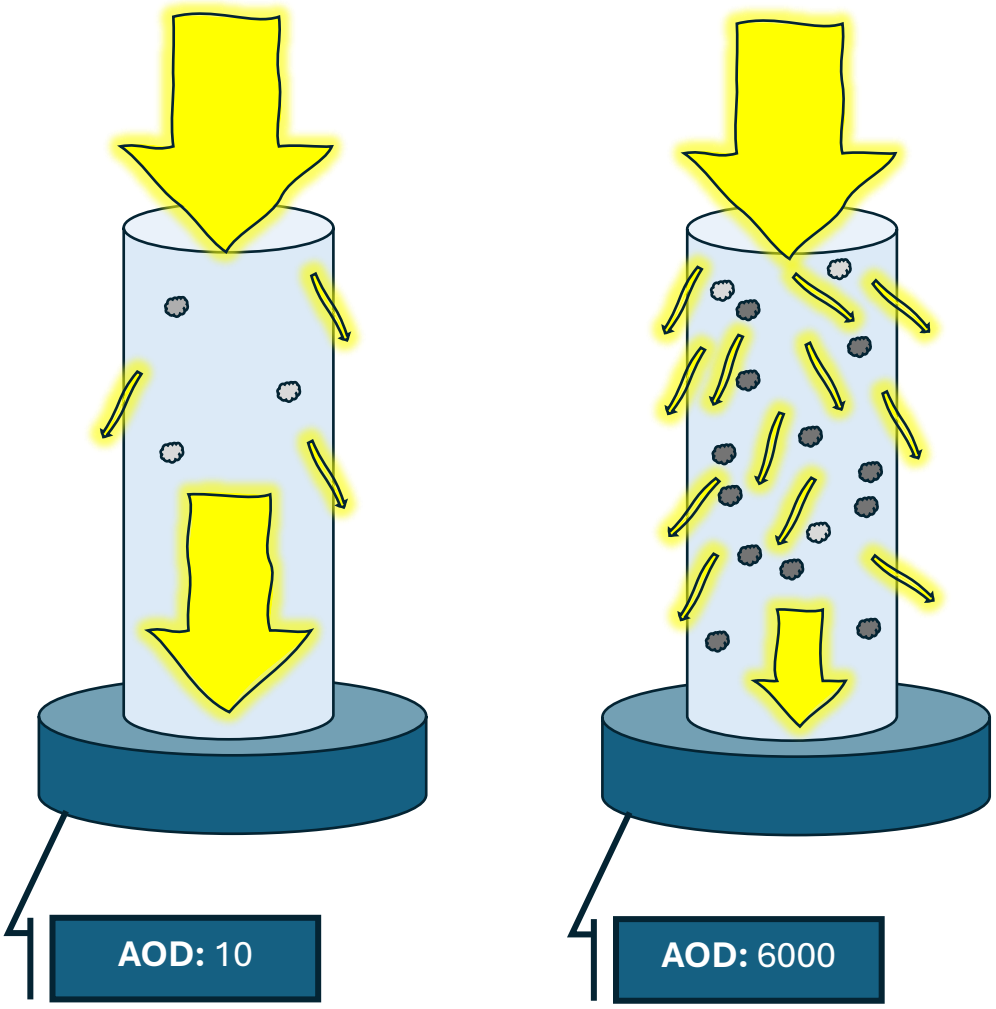
https://firms.modaps.eosdis.nasa.gov/map/

5 June to 23 July 2020, Tucson, Arizona

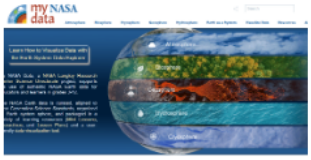
Aerosol Optical Depth (AOD)

Aerosol Optical Depth (AOD) is a value calculated via satellite measurements of aerosols (e.g., urban haze, smoke particles, desert dust) distributed within a column of air from Earth's surface to the top of the atmosphere.

1. **Column with High Aerosol Concentration**: This column contains a significant number of particles, leading to higher light scattering and absorption. As a result, less direct sunlight reaches the ground, corresponding to a higher AOD value.
2. **Column with Low Aerosol Concentration**: This column has fewer particles, resulting in minimal scattering and absorption of sunlight. Consequently, more direct sunlight reaches the ground, corresponding to a lower AOD value.



The data were collected by NASA's Terra satellite and accessed through the myNASAdata website.



my NASA data

My NASA Data is an educational platform designed to make NASA's Earth science data accessible to students, educators, and lifelong learners. The website offers a wealth of resources, including interactive tools, lesson plans, and real-world data sets, to support the...
read more

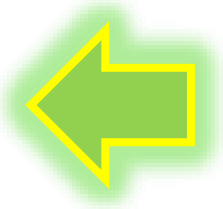
TERRA

Terra is a key NASA mission dedicated to studying Earth's systems from space, providing vital data for understanding our planet's climate, environment, and natural processes. Launched in 1999, the Terra satellite carries a suite of advanced scientific instruments that...
read more

Fortunately for us, someone has already gone to the trouble of plucking the data from the Internet for us.

Download the data and save the file on your computer (for later use):

Data.csv



CODAP gets the data talking

Visit the following site:

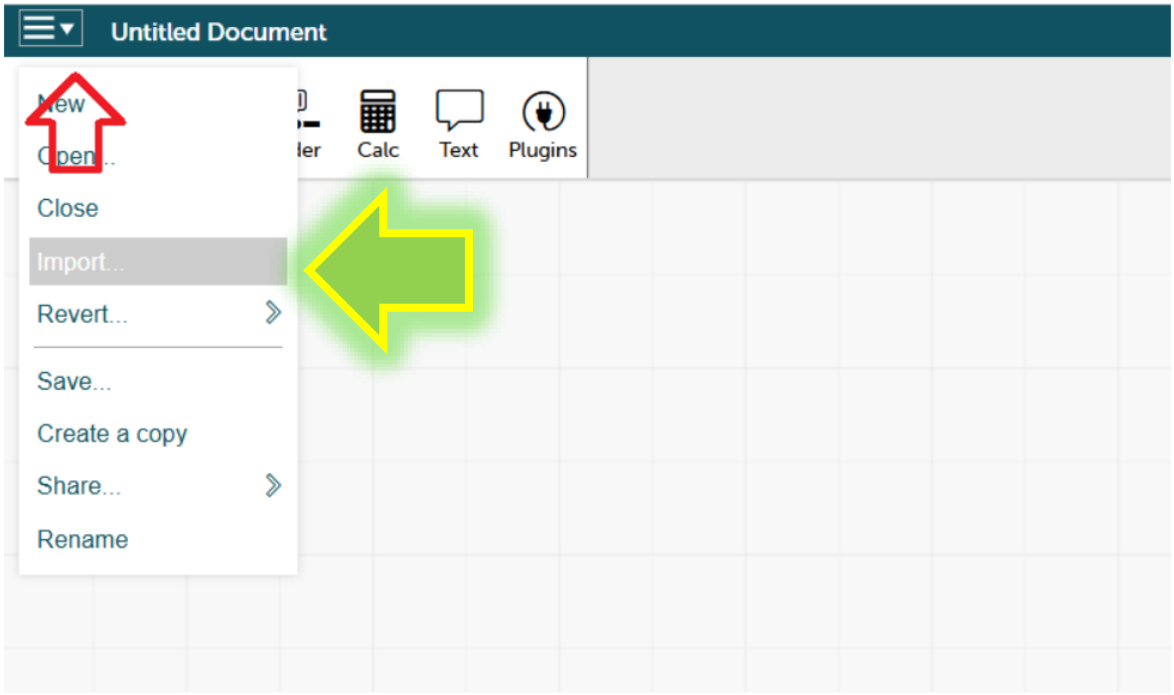
> Common Online Data Analysis Platform (CODAP)

https://codap.concord.org/

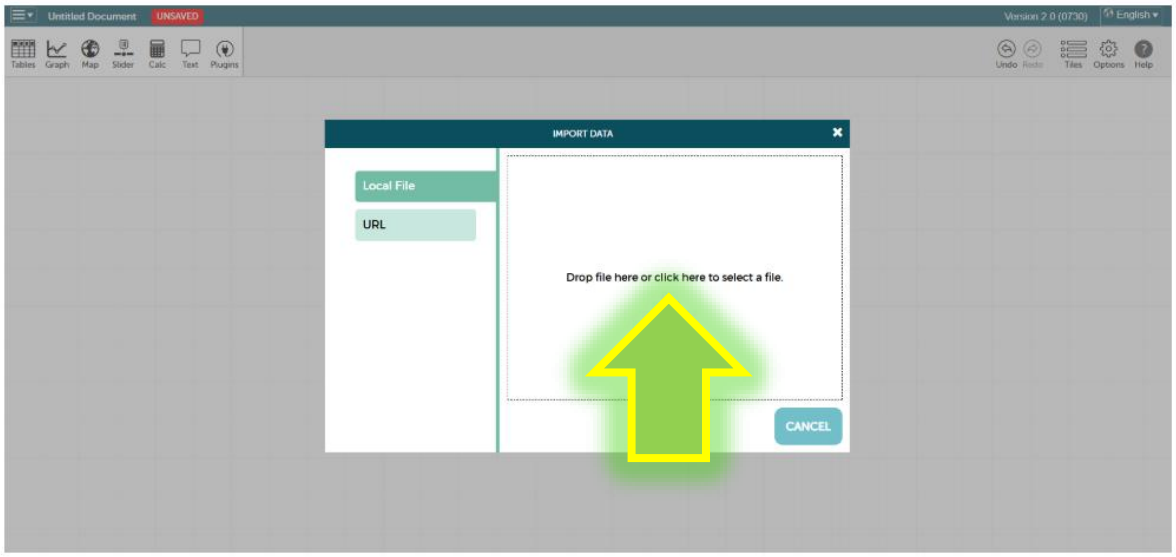
Launch CODAP ...

CREATE NEW DOCUMENT

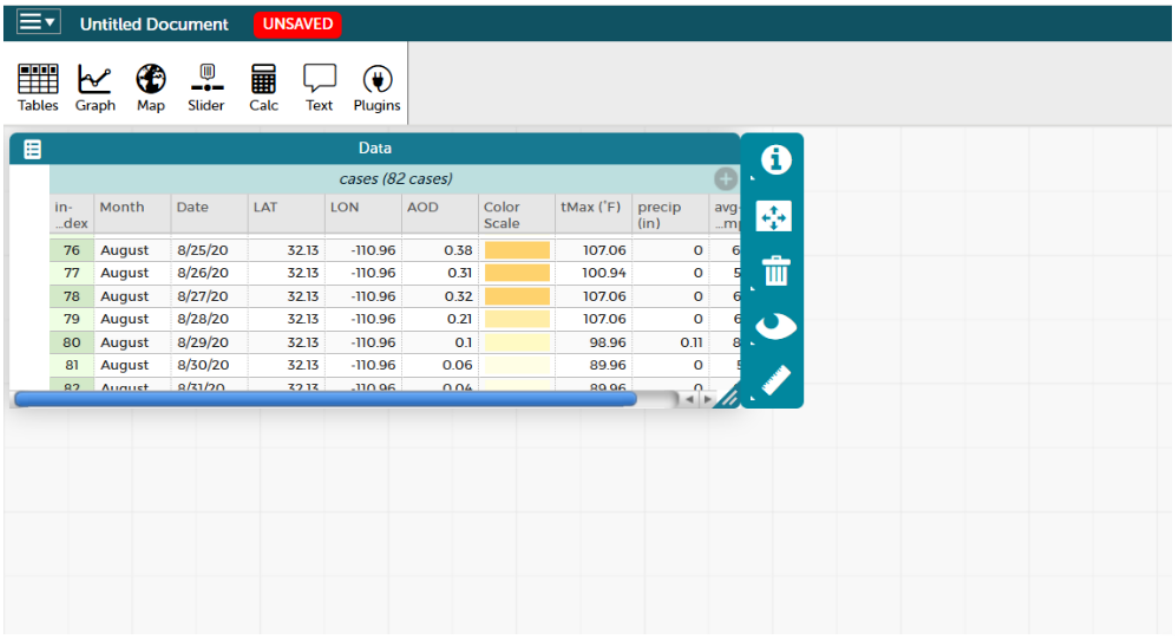
Drop-down menu in the top left-hand corner and then select ‘Import ...’.



We select the data file previously downloaded ...

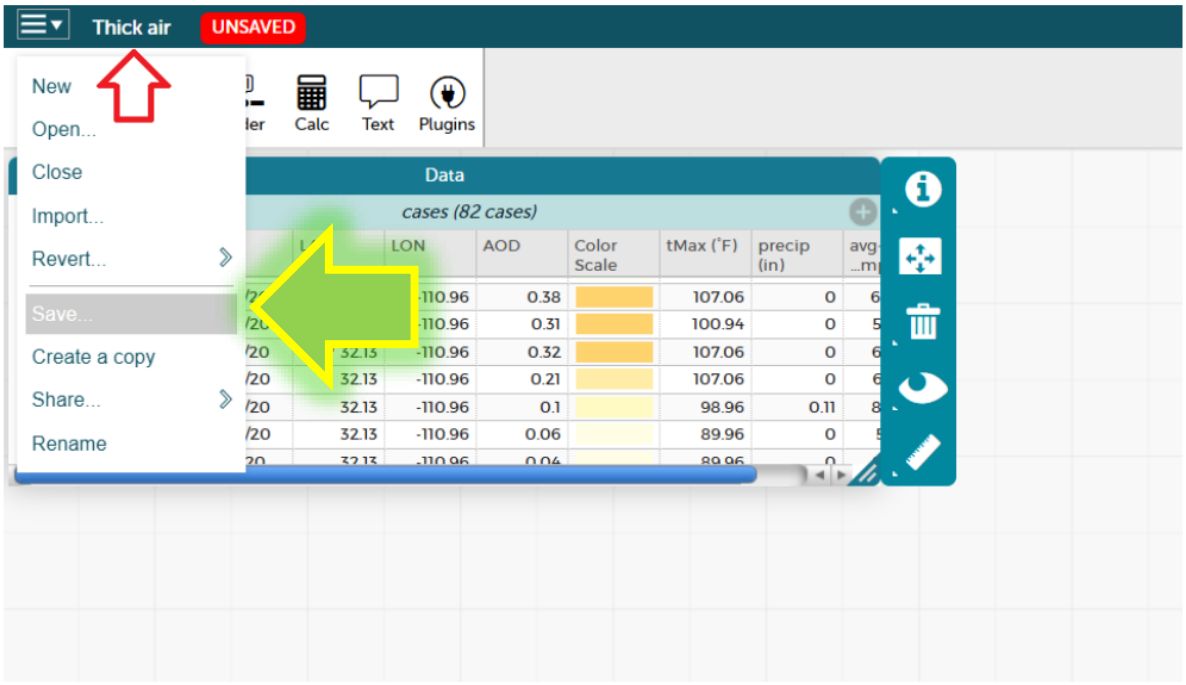


If everything went well, the data sets will now appear in the programme:



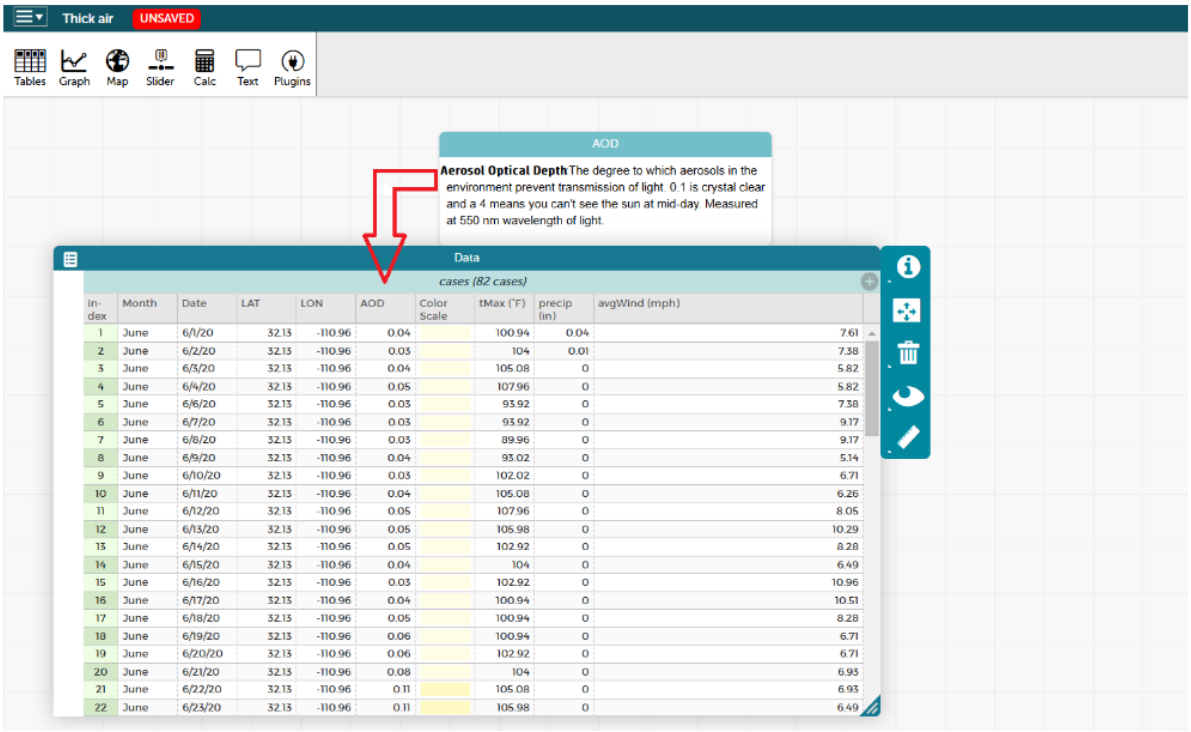
in-dex	Month	Date	LAT	LON	AOD	Color Scale	tMax (°F)	precip (in)	avg
76	August	8/25/20	32.13	-110.96	0.38		107.06	0	6
77	August	8/26/20	32.13	-110.96	0.31		100.94	0	5
78	August	8/27/20	32.13	-110.96	0.32		107.06	0	6
79	August	8/28/20	32.13	-110.96	0.21		107.06	0	6
80	August	8/29/20	32.13	-110.96	0.1		98.96	0.11	8
81	August	8/30/20	32.13	-110.96	0.06		89.96	0	5
82	August	8/31/20	32.13	-110.96	0.04		89.96	0	

Now it is time to give our document a name and save it:



Get to know your data

We have 82 data entries. A wide range of information is entered for each data entry. If you hold the mouse over a data field, detailed information is displayed:



index	Month	Date	LAT	LON	AOD	Color Scale	tMax (°F)	precip (in)	avgWind (mph)
1	June	6/1/20	32.13	-110.96	0.04		100.94	0.04	7.61
2	June	6/2/20	32.13	-110.96	0.03		104	0.01	7.38
3	June	6/3/20	32.13	-110.96	0.04		105.08	0	5.82
4	June	6/4/20	32.13	-110.96	0.05		107.96	0	5.82
5	June	6/6/20	32.13	-110.96	0.03		93.92	0	7.38
6	June	6/7/20	32.13	-110.96	0.03		93.92	0	9.17
7	June	6/8/20	32.13	-110.96	0.03		89.96	0	9.17
8	June	6/9/20	32.13	-110.96	0.04		93.02	0	5.14
9	June	6/10/20	32.13	-110.96	0.03		102.02	0	6.71
10	June	6/11/20	32.13	-110.96	0.04		105.08	0	6.26
11	June	6/12/20	32.13	-110.96	0.05		107.96	0	8.05
12	June	6/13/20	32.13	-110.96	0.05		105.98	0	10.29
13	June	6/14/20	32.13	-110.96	0.05		102.92	0	8.28
14	June	6/15/20	32.13	-110.96	0.04		104	0	6.49
15	June	6/16/20	32.13	-110.96	0.03		102.92	0	10.96
16	June	6/17/20	32.13	-110.96	0.04		100.94	0	10.51
17	June	6/18/20	32.13	-110.96	0.05		100.94	0	8.28
18	June	6/19/20	32.13	-110.96	0.06		100.94	0	6.71
19	June	6/20/20	32.13	-110.96	0.06		102.92	0	6.71
20	June	6/21/20	32.13	-110.96	0.08		104	0	6.93
21	June	6/22/20	32.13	-110.96	0.11		105.08	0	6.93
22	June	6/23/20	32.13	-110.96	0.11		105.98	0	6.49

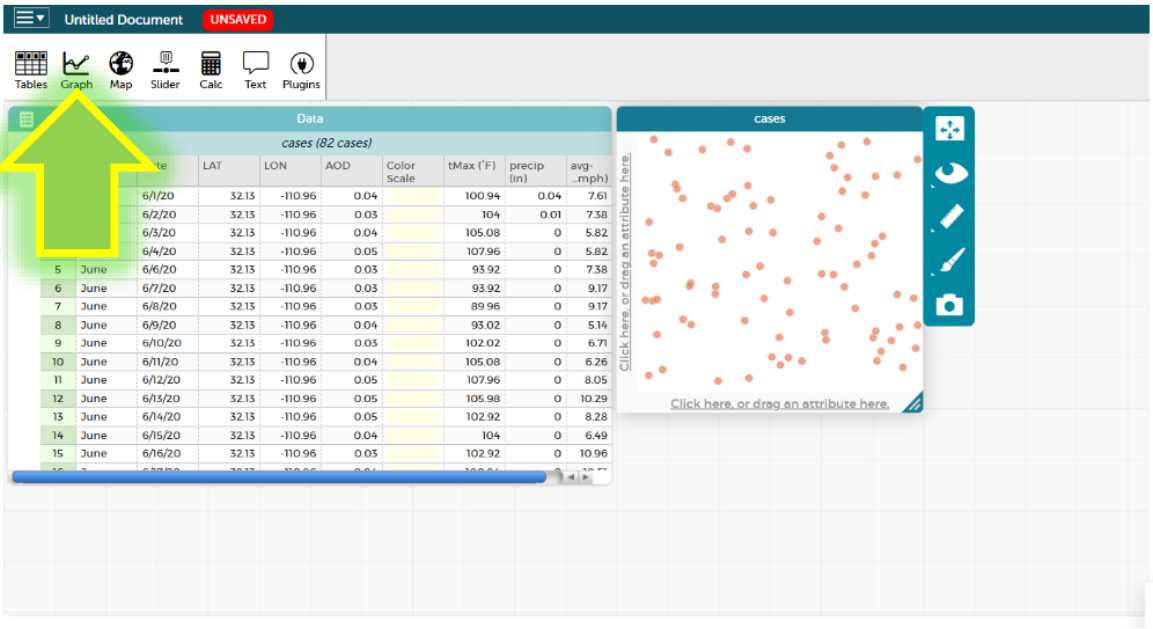
You will find a total of nine attributes (columns) for each of the 82 data entries (rows). An attribute can be numerical or categorical.

Aerosol optical depth (AOD) is calculated from satellite measurements of changes in the reflections and absorption of visible and infrared light due to particles in the atmosphere.

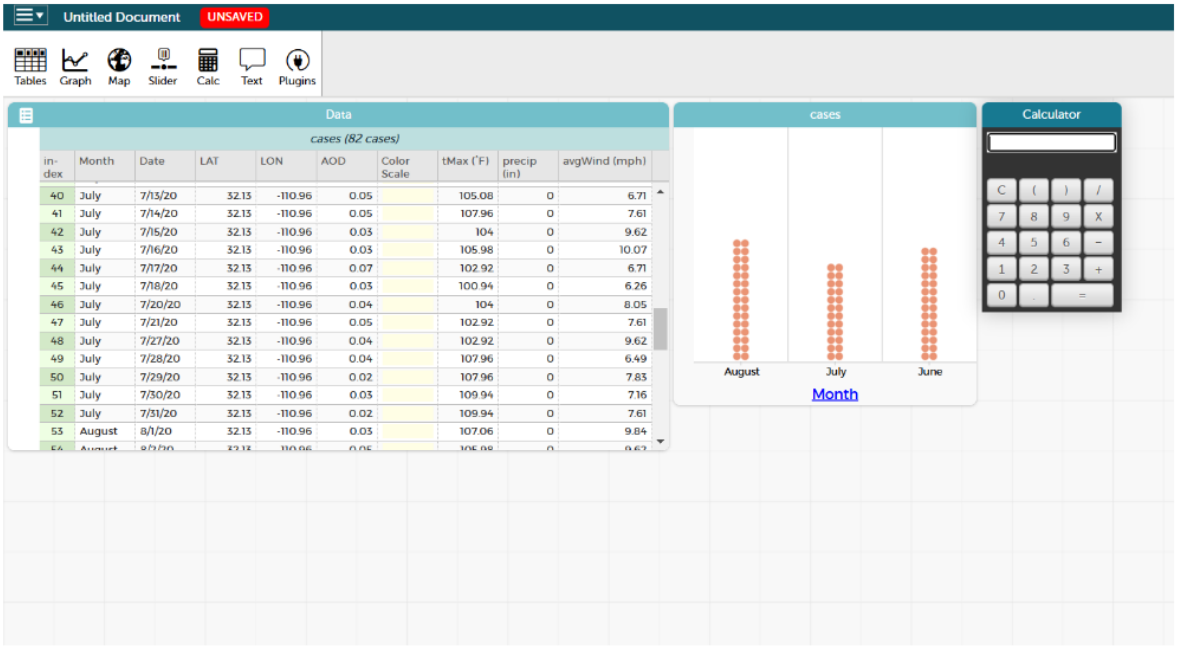
The data for AOD, temperature and other attributes were selected for a three-month period during which a large forest fire, known as the Bighorn Fire, occurred near Tucson, Arizona.

Juggling with data

The ‘Graph’ menu item initially shows us the entire, unorganised ‘data cloud’.

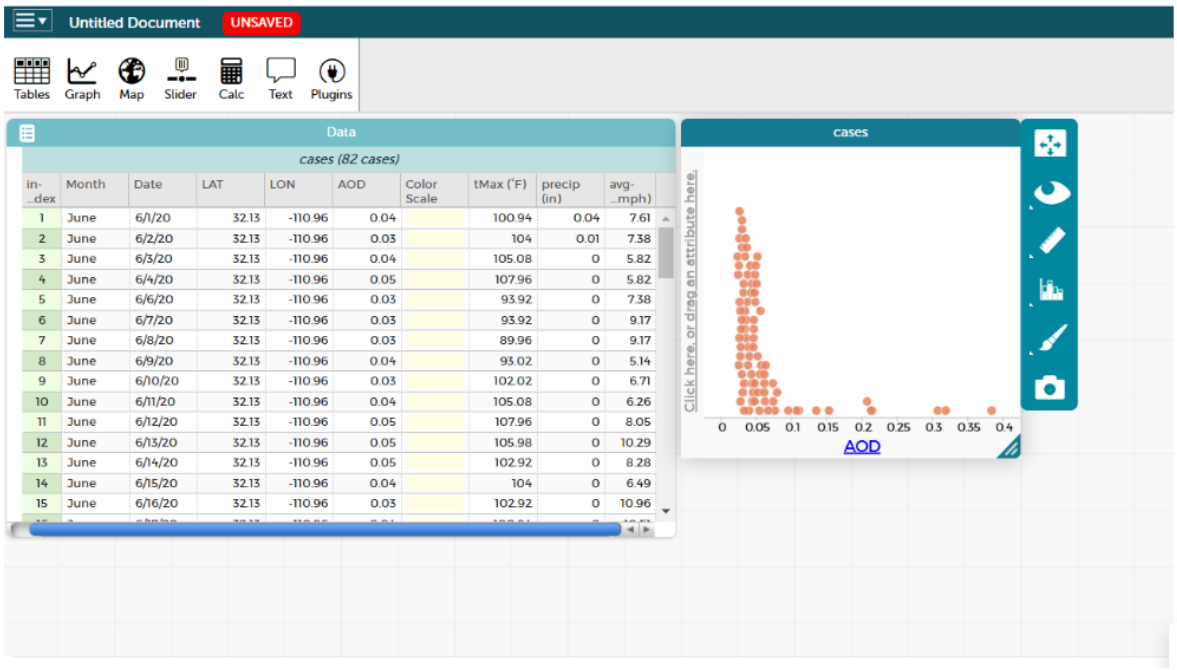


If we drag the 'Month' attribute onto the horizontal axis with the mouse, our pile of data already looks tidier.



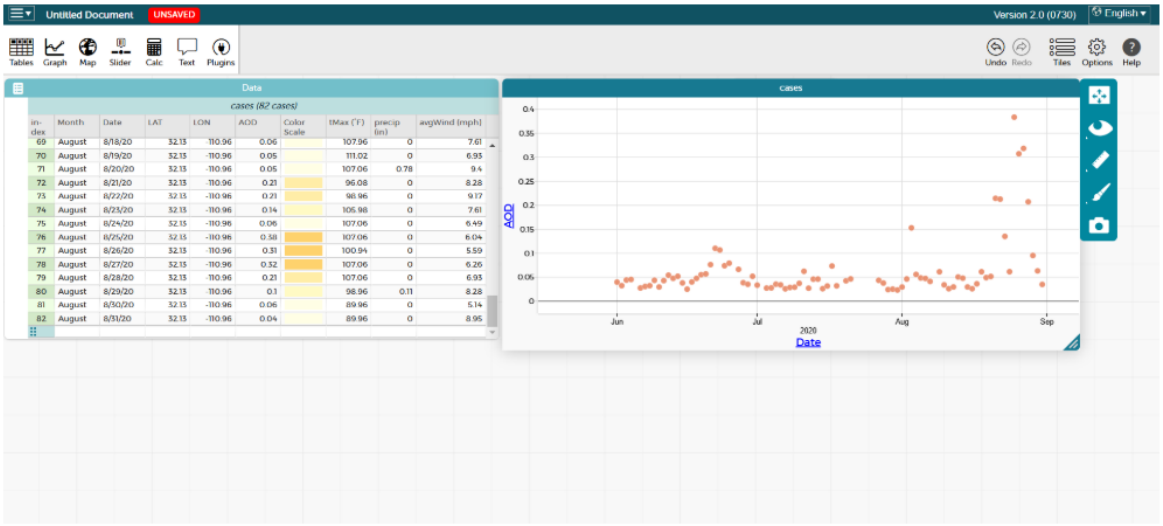
index	Month	Date	LAT	LON	AOD	Color Scale	tMax (°F)	precip (in)	avgWind (mph)
40	July	7/13/20	32.13	-110.96	0.05		105.08	0	6.71
41	July	7/14/20	32.13	-110.96	0.05		107.96	0	7.61
42	July	7/15/20	32.13	-110.96	0.03		104	0	9.62
43	July	7/16/20	32.13	-110.96	0.03		105.98	0	10.07
44	July	7/17/20	32.13	-110.96	0.07		102.92	0	6.71
45	July	7/18/20	32.13	-110.96	0.03		100.94	0	6.26
46	July	7/20/20	32.13	-110.96	0.04		104	0	8.05
47	July	7/21/20	32.13	-110.96	0.05		102.92	0	7.61
48	July	7/27/20	32.13	-110.96	0.04		102.92	0	9.62
49	July	7/28/20	32.13	-110.96	0.04		107.96	0	6.49
50	July	7/29/20	32.13	-110.96	0.02		107.96	0	7.83
51	July	7/30/20	32.13	-110.96	0.03		109.94	0	7.16
52	July	7/31/20	32.13	-110.96	0.02		109.94	0	7.61
53	August	8/1/20	32.13	-110.96	0.03		107.06	0	9.84

Let’s drag another attribute onto the horizontal axis:

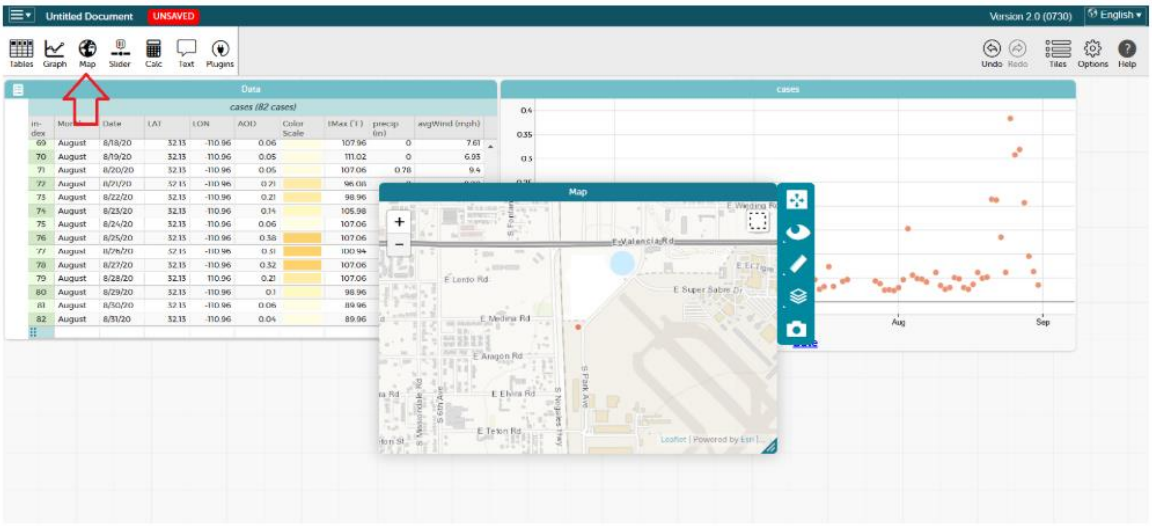


in-..dex	Month	Date	LAT	LON	AOD	Color Scale	tMax (°F)	precip (in)	avg-..mph)
1	June	6/1/20	32.13	-110.96	0.04		100.94	0.04	7.61
2	June	6/2/20	32.13	-110.96	0.03		104	0.01	7.38
3	June	6/3/20	32.13	-110.96	0.04		105.08	0	5.82
4	June	6/4/20	32.13	-110.96	0.05		107.96	0	5.82
5	June	6/6/20	32.13	-110.96	0.03		93.92	0	7.38
6	June	6/7/20	32.13	-110.96	0.03		93.92	0	9.17
7	June	6/8/20	32.13	-110.96	0.03		89.96	0	9.17
8	June	6/9/20	32.13	-110.96	0.04		93.02	0	5.14
9	June	6/10/20	32.13	-110.96	0.03		102.02	0	6.71
10	June	6/11/20	32.13	-110.96	0.04		105.08	0	6.26
11	June	6/12/20	32.13	-110.96	0.05		107.96	0	8.05
12	June	6/13/20	32.13	-110.96	0.05		105.98	0	10.29
13	June	6/14/20	32.13	-110.96	0.05		102.92	0	8.28
14	June	6/15/20	32.13	-110.96	0.04		104	0	6.49
15	June	6/16/20	32.13	-110.96	0.03		102.92	0	10.96

Why shouldn’t we also assign an attribute to the vertical axis? We have enough attributes!

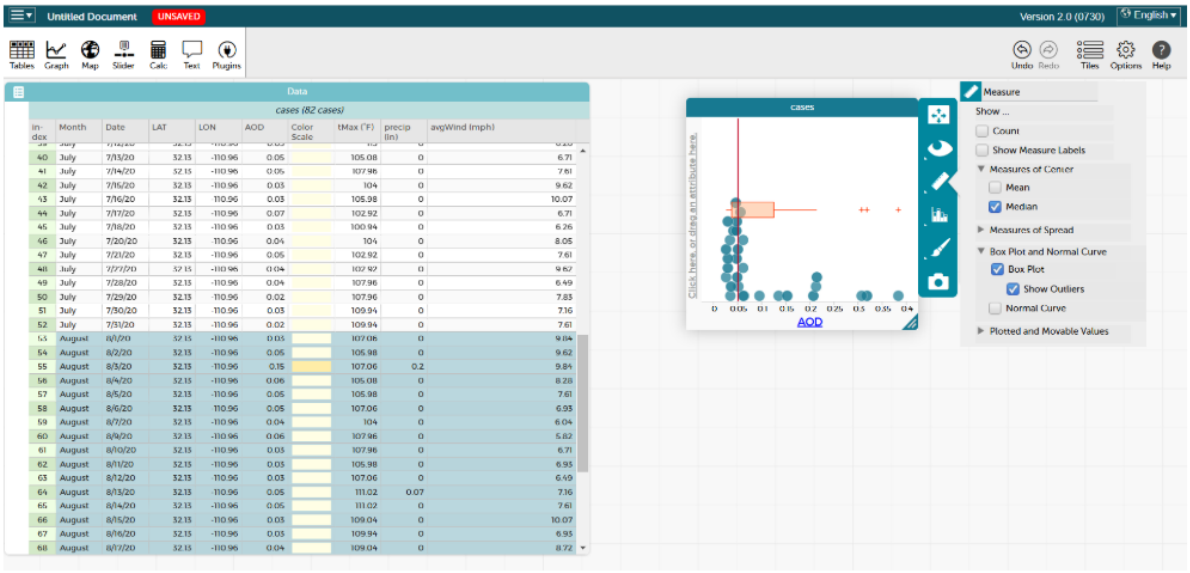


Maybe we'll also try something from the menu bar. How about 'Map'?



Looks like a lonely data point at the airport. Let's check this with Google maps!

With a little perseverance, we can now start analysing the data. We can mark individual data areas and use such wonderful things as Mean, Median or BoxPlot to find out what the forest fire has done to the data.



I wanted to find out about the optical depth categories of aerosols at 550 nm. For example, what does it mean if the value is below, say, 0.05. Can you provide me with a general categorization table for this?

Aerosol Optical Depth (AOD) at 550 nm quantifies the degree to which aerosols in the atmosphere prevent the transmission of sunlight by absorption or scattering. AOD values range from 0 (completely clear sky) to 1 or more (very hazy conditions).

While there isn't a universally standardized categorization for AOD values, a general classification is as follows:

- **AOD < 0.1**: Clear conditions with minimal aerosol presence.
- **0.1 ≤ AOD < 0.25**: Light aerosol loading, typical of background conditions.
- **0.25 ≤ AOD < 0.4**: Moderate aerosol presence, indicating noticeable haze.
- **AOD ≥ 0.4**: High aerosol loading, leading to significant haze and reduced visibility.

Making sense of data

Note: This dataset is intended for students to find no effect of the wildfire on the AOD for Tucson, Arizona.

The AOD data here are not elevated due to the wildfire. This dataset can be used to help students understand that approaching a dataset with bias or expectation can influence what one finds in the data. There are some higher AOD values toward the end of the data, which students may assume represent the fire.

Deliberate Practice and …

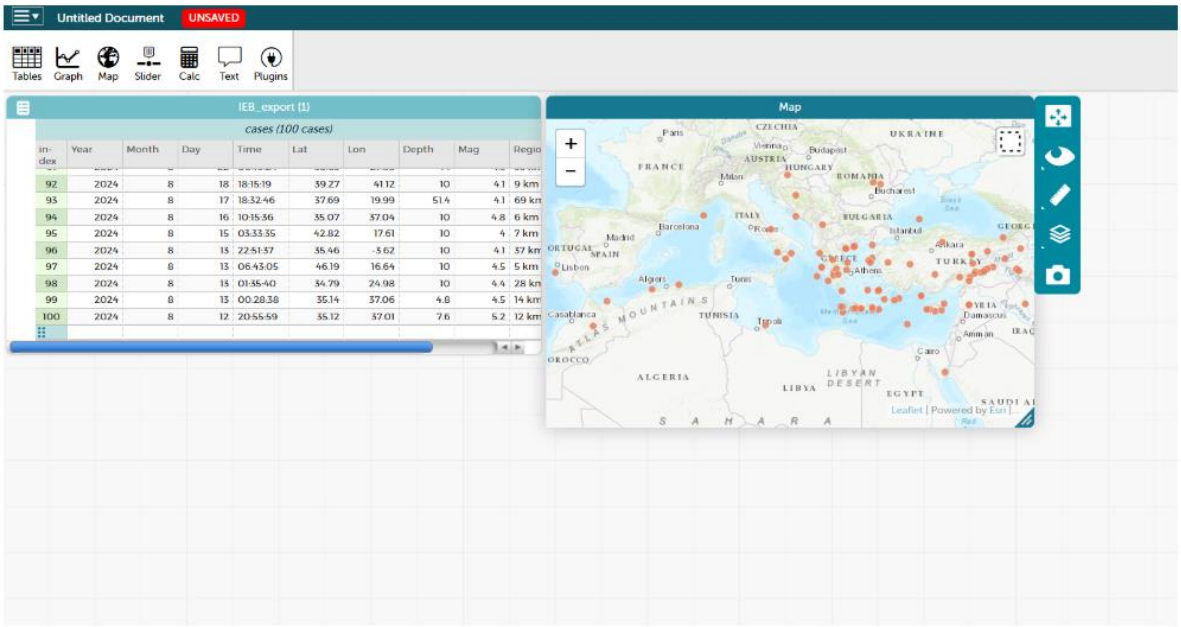
A Second example …

Earthquakes

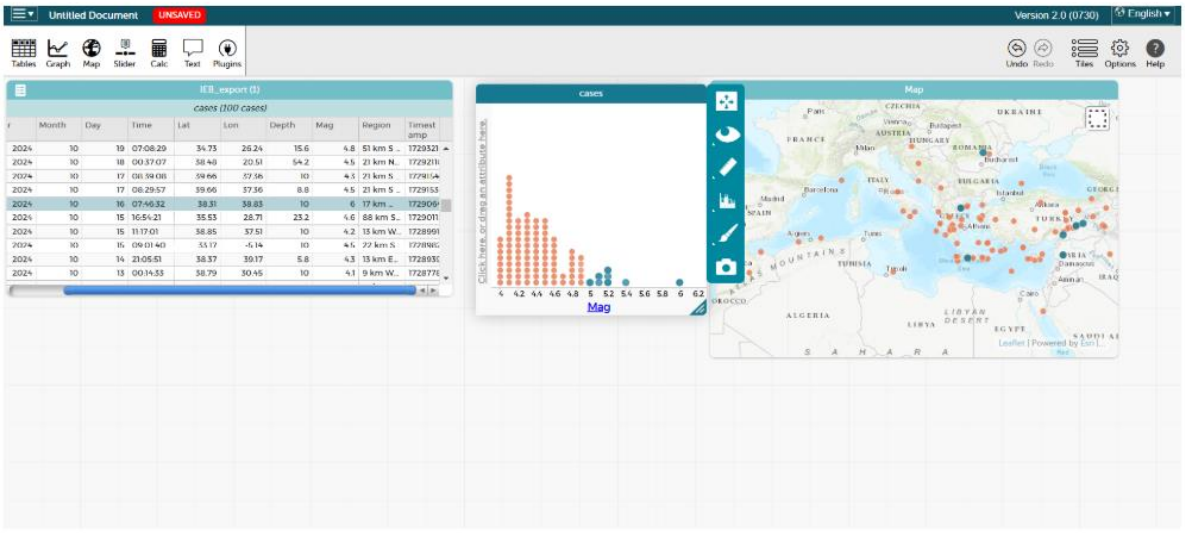
Data Set is here:

[earthquakes.csv]

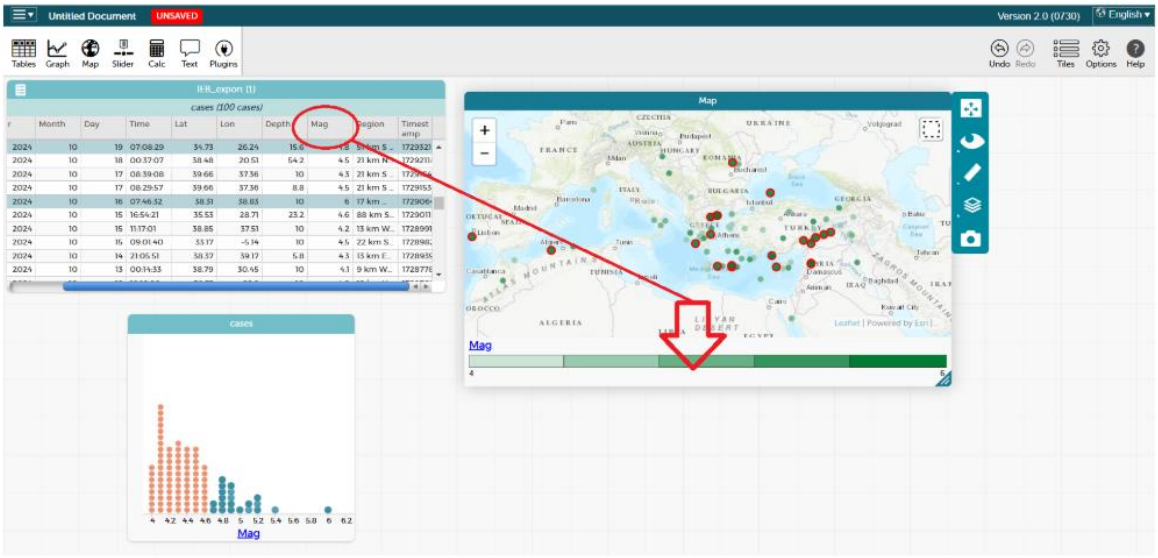
Firstly: Import data



Then: Organise the pile of data

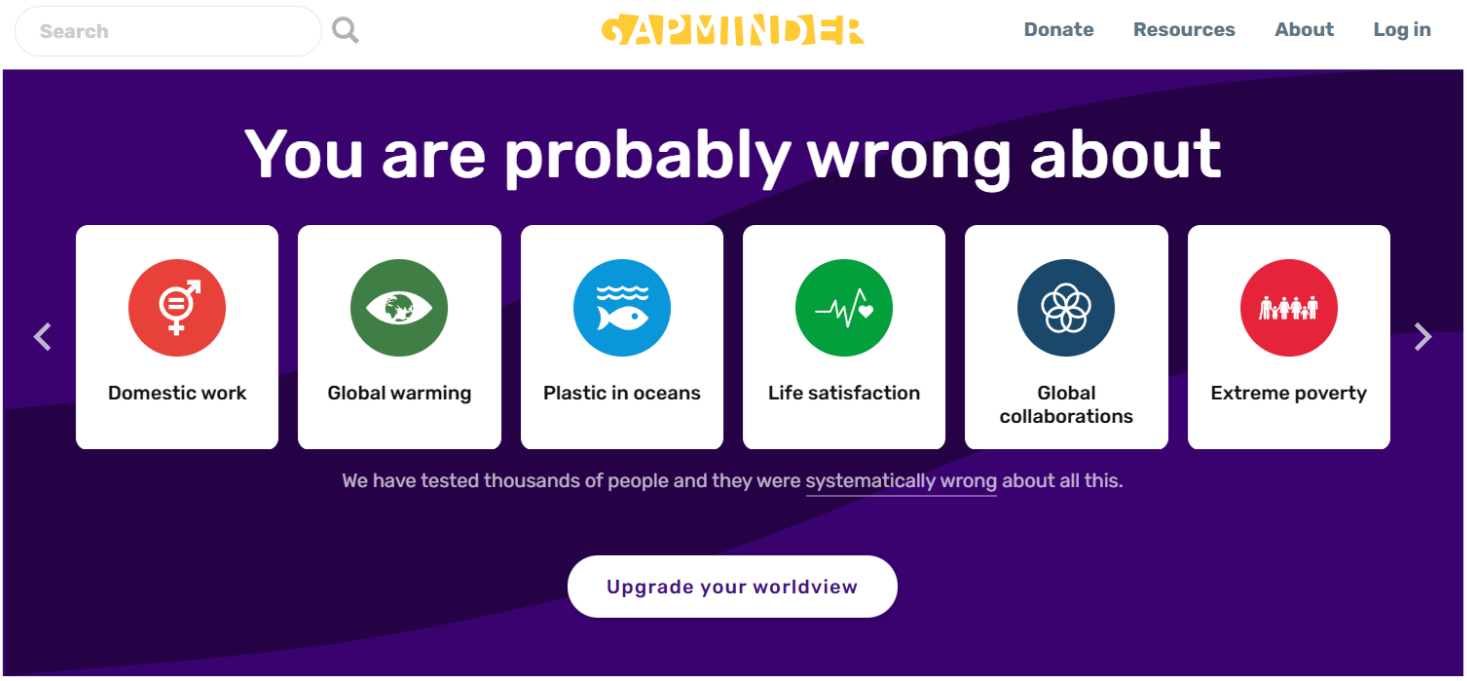


What happens when you drag the ‘Mag’ attribute to the Map with the mouse?



TRAINING

Get meaningful data from here ...



Gapminder is an independent educational non-profit fighting global misconceptions.

https://www.gapminder.org/

... and use CODAP to find meaningful mean value.