

Begeisterung für digitale Kompetenzen und Programmieren – mach Hacker School am Girls' Day!

Eine Kooperation der bundesweiten Koordinierungsstelle
Girls' Day – Mädchen-Zukunftstag
mit der Hacker School

HACKER SCHOOL

Inhalt

Hinweise zum Konzept

1. Begrüßung mit „Eisbrecher-Fragen“ (15 Min.)
2. Unternehmenspräsentation – virtuelle Führung und Interview mit Vorbildern (15 Min.)
3. Impuls – Digitale Kompetenzen (15 Min.)

Pause

4. Praxisteil Programmieren: „Klima mit Python“ (3 Std.)
5. Feedback und Verabschiedung (15 Min.)

Links zu weiteren Projekten und Informationen

Anhang: Digitale Kompetenzen nach DigComp (EU)

Hinweise zum Konzept

- Vorschlag für die Dauer des Kurses: 4 Stunden
- Die Anzahl der möglichen Teilnehmerinnen richtet sich nach der Anzahl der zur Verfügung stehenden Mitarbeiter*innen aus dem Unternehmen, die den Praxisteil begleiten. Unsere Empfehlung: 2:10 – zwei Mitarbeiter*innen begleiten eine Gruppe von zehn Schülerinnen.
- Tragen Sie die Veranstaltung ins [Girls' Day-Radar](#) ein, dies bietet viele Vorteile:
 - Sichtbarkeit: viele Schülerinnen nutzen das Radar für ihre Suche
 - Teilnehmerinnen-Management: Online-Anmeldung für Mädchen; Anmelde Listen in Excel erstellen; personalisierte E-Mails an Teilnehmerinnen versenden; vorgefertigte Teilnahmebestätigungen und Fotofreigabeerklärung verschicken
 - Über Ihr persönliches Konto „Mein Girls' Day“ können Sie Ihren Eintrag bearbeiten und Angebote mit verschiedenen Ansprechpersonen verwalten.
- Binden Sie die Schülerinnen schon vor dem Girls' Day mit ein, zum Beispiel:
 - Richten Sie ein Padlet ([padlet.com](#)) ein und verschicken Sie den Link, damit die Schülerinnen ein Foto (zum Beispiel von sich selbst, einem Haustier o.ä.) hochladen und sich kurz vorstellen können. Hier können die Schülerinnen vorab auch schon Fragen einstellen, die am Aktionstag von den Vorbildern (Role Models) im Interview beantwortet werden.
 - Bestellen Sie im [Materialcenter](#) kostenfreie Girls' Day-Aktionsmaterialien, um Ihr Angebot zu bewerben. Verschicken Sie auch Girls' Day-Give-Aways, die Sie kostenpflichtig ebenfalls im Materialcenter bestellen können.
 - Verschicken Sie Fotos von Vorbereitungen im Unternehmen, z. B. wie kleine „Pausenpakete“ (mit Müsliriegeln o.ä.) für eine gemeinsame Mittagspause gepackt werden.

Beispiel: [padlet.com](https://www.padlet.com)



Ihr Rezept für ein gelungenes digitales Angebot!

Einen Leitfaden mit weiteren Informationen und Tipps zur Planung und Durchführung digitaler Angebote finden Sie [hier](#).

Orientierung

Inspiration

Interaktion

Unterhaltung

1. Begrüßung mit „Eisbrecher-Fragen“

Gestalten Sie die Begrüßung der Schülerinnen interaktiv und bereiten Sie Fragen mit verschiedenen Antwortmöglichkeiten vor. Dies können **Schätzfragen, Fragen zu Erwartungen an den Kurs oder Fragen zur Selbsteinschätzung** sein. Für die Darstellung der Antworten empfehlen wir Mentimeter ([mentimeter.com](https://www.mentimeter.com)), eine kostenfreie und webbasierte Anwendung, die Abstimmungen in Echtzeit darstellt.

Beispiele für „Eisbrecher-Fragen“:

- Seit wann gibt es den Girls' Day – Mädchen-Zukunftstag?

Antwortmöglichkeiten: 1992 – 1995 – **2001** – 2010

- Wie viele Programmiersprachen gibt es?

Antwortmöglichkeiten: 1 – 6 – 53 – **unbestimmt viele**

- Worauf freust du dich heute am meisten?

Antwortmöglichkeiten: Rundgang im Unternehmen – Interview mit XY – Programmieren

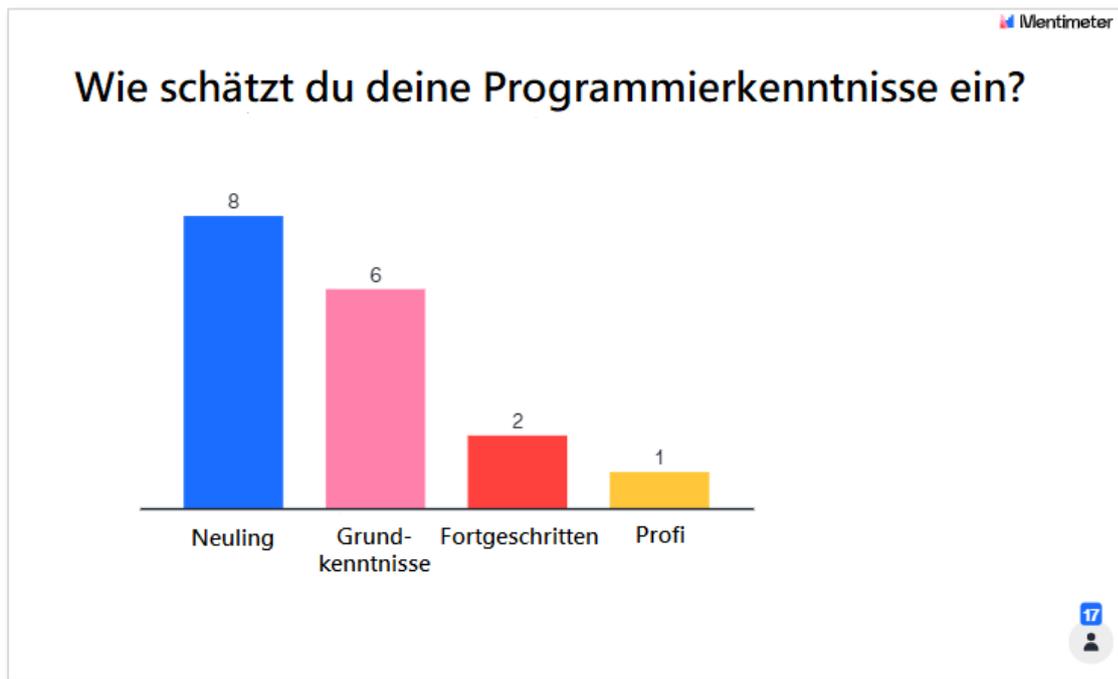
- Wer war die erste Programmiererin?

Antwortmöglichkeiten: Marie Curie – Dorothy Vaughan – **Ada Lovelace** – Katherine Johnson

- Wie schätzt du deine Programmierkenntnisse ein?

Antwortmöglichkeiten: Neuling – Grundkenntnisse – Fortgeschritten – Profi

Beispiel: [mentimeter.com](https://www.mentimeter.com)



2. Unternehmenspräsentation – virtuelle Führung und Interview mit Vorbildern

15 Min.

Stellen Sie Ihr Team und Ihr Unternehmen vor, zum Beispiel mit einer **virtuellen Smartphone-Führung**.

Stellen Sie dabei Bereiche heraus, die für die Schülerinnen relevant und interessant sind – hier steht das **Erleben** im Vordergrund. Binden Sie Auszubildende und insbesondere junge Mitarbeiterinnen als **Vorbilder** (Role Models) mit ein.

- Wie sieht es in Ihrem Unternehmen aus?
- Welche Aufgaben hat das Team, das vorgestellt wird? Wie gestaltet sich die Zusammenarbeit?
- Wie ist die kollegiale Atmosphäre?

Inspiration
Zeigen Sie
Mitarbeiterinnen
in konkreten
Arbeitssituationen

Mit der Smartphone-Führung erhalten die Schülerinnen einen ersten Eindruck und grundlegende Informationen über das Unternehmen.

In einem **Interview mit einer Auszubildenden oder jungen Mitarbeiterin** (Role Model) erhalten die Schülerinnen ganz konkrete Informationen zu einem bestimmten Girls' Day-Beruf, zum Beispiel:

- Einstiegs- und Aufstiegsmöglichkeiten
- konkrete Tätigkeiten und Arbeitssituationen
- Gehaltsperspektiven
- persönliche Erfahrungen in Ausbildung, Studium und Beruf
- Infos zu individuellen Fragen der Schülerinnen (siehe ggf. Fragen auf dem Padlet)

3. Impuls – Digitale Kompetenzen

15 Min.

Digitale Kompetenz deckt ein breites Spektrum an Kenntnissen, Fähigkeiten und Strategien ab, die in ihrer Gesamtheit von grundlegender Bedeutung für gesellschaftliche Teilhabe sind – sowohl im Beruf als auch im Privaten.

Programmierkenntnisse beschreiben dabei nur einen Bereich digitaler Kompetenz.

Digitalkompetenzen sind die Schlüsselkompetenzen des 21. Jahrhunderts für alle Menschen in unserer Gesellschaft.

Die EU-Kommission identifiziert im Rahmen von „[DigComp 2.0: The Digital Competence Framework for Citizens](#)“ fünf Bereiche digitaler Kompetenz, in denen alle Menschen grundlegende Kenntnisse haben sollten:

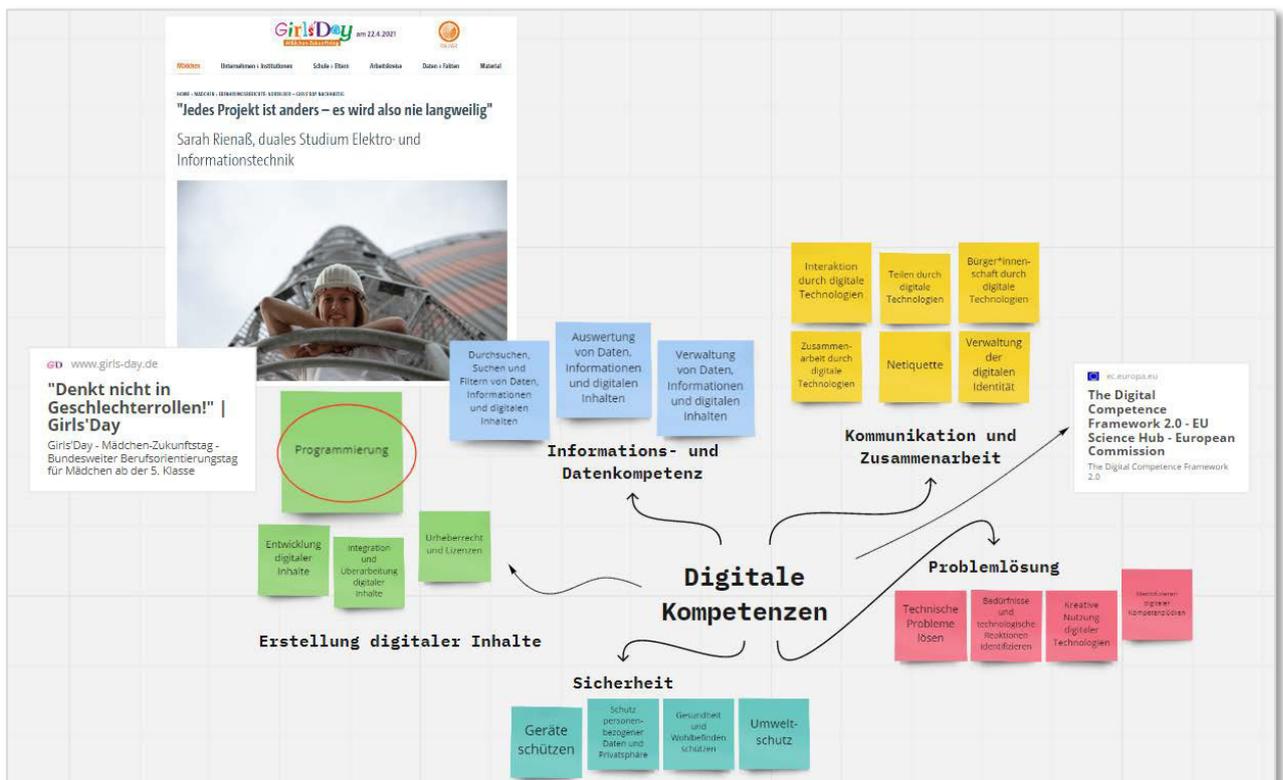
1. Informations- und Datenkompetenz
2. Kommunikation und Zusammenarbeit
3. Erstellung digitaler Inhalte
4. Sicherheit
5. Problemlösung

Tool-Empfehlung: Miro-Board

Führen Sie durch die Vielfalt und Bereiche digitaler Kompetenz und stellen Sie dabei den konkreten Bezug zu Berufen, Tätigkeiten und Arbeitssituationen in Ihrem Unternehmen heraus.

Die Nutzung des Miro-Boards (miro.com) ermöglicht, komplexe Inhalte grafisch anschaulich und Schritt für Schritt zu erläutern. Darüber hinaus können Sie den theoretischen Teil mit eingebetteten Fotos, Videoclips oder Links informativ und unterhaltsam gestalten.

Beispiel: miro.com



4. Praxisteil Programmieren: „Klima mit Python“

3 Std.

Das 3-stündige Kurskonzept wurde von der Hacker School entwickelt und führt in die Grundlagen der Programmiersprache Python ein.

Beginnen Sie mit einer kurzen Vorstellungsrunde und teilen Sie die Schülerinnen idealerweise nach Erfahrung in Gruppen ein. Die Kurseinheit endet mit einer Abschlussrunde, in der einzelne Ergebnisse in Kürze vorgestellt werden können.



© kompetenz

Hacker School @yourSchool Python

Genereller Ablauf

Hinweise zum Konzept

Lernziele

Technologie / Repl.it

Konzept

Begrüßung und generelle Motivation (~15 Min)

Vorstellungsrunde (Kleingruppe)

Programmieren (~ 3h + 30 min Pause) (in der Kleingruppe)

Hallo Welt!

Hallo <Name>!

If und Else

Datentypen und Funktionen

CO2 Input

Code lesen (in der Kleingruppe)

Projektarbeit

(Größeres) Beispielprojekt

Interne Abschlussrunde (Kleingruppe, ~5 Minuten)

Abschlussrunde (Plenum, ~20 min)

Genereller Ablauf

- Der Kurs beginnt mit der Begrüßung im Plenum. Ein Inspirer übernimmt die Einführung in das Thema und versucht sicherzustellen, dass die Technik bei den Meisten funktioniert. Danach werden die Kinder in Zweier-Teams eingeteilt.
- Die eingeteilten Teams werden in Breakout-Rooms (bei Zoom) eingeteilt. Wenn ihr in den Breakout Rooms Fragen habt, ruft gerne über Zoom um Hilfe. Die Inspirer wird dann zu euch in den Breakout-Room kommen.
- Vorstellungsrunde in den Gruppen. Stellt euch einmal vor und erklärt gerne, wie ihr zu der IT gekommen seid. Lasst die Kids sich vorstellen, inklusive Programmiererfahrung, Erwartungen und Motivation.
- Jetzt geht's ans Eingemachte: Programmieren! Mehr Informationen zu den Einheiten findet ihr unten.
- Versucht nach etwa der Halbzeit eine Pause von 30 Minuten einzulegen.
- Der Abschluss findet wieder im Plenum mit der gesamten Klasse statt. Hier stellt sich jeder Inspirer einmal kurz vor und erklärt, warum er*sie für IT brennt.

P.S.: Vergesst nicht, eure Pausen zu machen :)

Hinweise zum Konzept

Lernziele

- Es geht darum, die Kids zu begeistern. Unser Ziel ist es, dass sie sich nach den 3 Stunden Kurs noch 12 Stunden hinsetzen und weiter recherchieren und programmieren. Wie das bei euch konkret aussieht, ist euch überlassen. Ihr merkt, die Kids haben voll Lust mit euch eine halbe Stunde über Roboter zu reden? Dann macht das gerne :)
- Versucht eher, den Kids grundlegende Konzepte beizubringen (“Wie debugge ich ein Programm?”), als Ihnen diese eine Eigenheit von Python lang und breit zu erklären. Zeigt/erklärt ihnen auch gerne, wo das, was sie gerade lernen, im echten Leben zum Einsatz kommt.
- Lasst die Kids so viele Schritte wie möglich selbst gehen. Sie werden so am meisten lernen, und sind hoffentlich auch in der Lage, an das Gelernte nach Ende des Kurses anknüpfen zu können.
- Lasst sie dabei auch scheitern. Im Optimalfall versucht ihr dann mit ihnen gemeinsam zu ergründen, wieso ihr Ansatz nicht funktioniert hat - dadurch müssen sie ihre Gedanken kommunizieren, und lernen so mehr.
- Dieses Konzept ist unser Vorschlag, wie man Kinder und Jugendliche begeistern könnte. Seht es also mehr als groben Leitfaden und weniger als festen Lehrplan.

Technologie / Repl.it

- Wir arbeiten während dem Kurs mit dem Online-Editor Repl.it -> <https://repl.it>
- Wie genau die Kids sich einloggen und mit welchen Accounts sie arbeiten, erklärt der Inspirer am Anfang im Plenum.

Konzept

- Wir wollen, dass ihr alle etwas mitnehmen könnt. Uns ist es lieber, dass ihr Zusammenhänge verstanden habt, aber dafür weniger. (Wird trotzdem vorkommen, dass euch Dinge überfordern, aber das ist normal. Behaltet das trotzdem im Hinterkopf)
 - Vordergründig bringen wir euch heute Programmieren bei, aber eigentlich geht es uns darum, euch mit Menschen zusammenzubringen, die das richtige Mindset haben. Damit ihr von Ihnen lernen könnt. Wenn ihr inhaltlich überhaupt nichts verstehen solltet, aber dafür die Fehlerkultur mitnehmt, sind wir happy.
 - A propos Fehlerkultur: Wenn ihr etwas nicht versteht - **fragt nach**. Das hier soll alles werden, bloß kein Frontalunterricht, in dem ihr Dinge schlucken, verdauen, und nachher wieder auskotzen sollt. Uns geht es darum, dass ihr Spaß habt, und vor allem: etwas lernt. Und gerade dem stehen so ein paar Dinge im Weg, wie: Schon beim ersten kleinen Fitzelchen Syntax zu scheitern.
Also: Wenn ihr jetzt gleich euer Programm nicht funktioniert, was macht ihr dann? Richtig. **Fragt nach**.
- grober inhaltlicher Überblick
- zeitlicher Überblick
- Einmal kurz die Technologie durchgehen (repl.it)

Vorstellungsrunde

- Stellt euch selber vor. (2-3 Sätze)
 - Was habt ihr gemacht, als ihr so alt wart wie die Teilnehmenden?
- Lasst die Teilnehmenden sich vorstellen.
 - Wie heißt du?
 - Wieso bist du hier?
 - Hast du schonmal programmiert?
 - Was erhoffst du dir von diesem Kurs?
- Wichtig: achtet darauf, dass die Kinder sich nur mit Vornamen vorstellen (Datenschutz :)

Programmieren (~ 3h + 30 min Pause) (in der Kleingruppe)

Hallo Welt!

- Teilt euren Bildschirm und programmiert Hallo Welt.
- Erklärt gerne Zeichen für Zeichen
 - Warum kommen da jetzt Klammern hin? - Und welche?
 - Was bedeuten die `""` dort?
 - Was ist eine Konsole?
 - Wie führe ich das Programm aus? (Ggf. auch auf mobilen Endgeräten)
- Lasst die Kinder, nachdem ihr fertig seid, selber ausprobieren eigene Texte auf der Konsole auszugeben.

Was sie lernen

- print()
- Strings
- indirekt Funktionsaufrufe (muss aber noch nicht erklärt werden, darauf könnt ihr aber später zurückkommen)
- Was die Konsole macht/ist
- Wie ich ein Programm starte

Code

```
print("Hallo Welt!")
```

Hallo <Name>!

- Teilt euren Bildschirm und fügt den Code unten zu dem bereits bestehendem Hallo Welt Programm hinzu.
- Lasst die Kinder, nachdem ihr fertig seid, selber ausprobieren eigene kreative Fragen und angepasste Antworten auf der Konsole auszugeben.

Was sie lernen

- Variablendeklaration mit `<Variablenname> = <Wert>`
- mit `input("")` eine Frage stellen
- Wieder auf Variablen zugreifen

Code

```
name = input("Wie heißt du? ")  
print("Hallo " + name)
```

If und Else

- Teilt euren Bildschirm und ersetzt das print() aus "Hallo, <Name>!" mit dem Code unten.

Was sie lernen

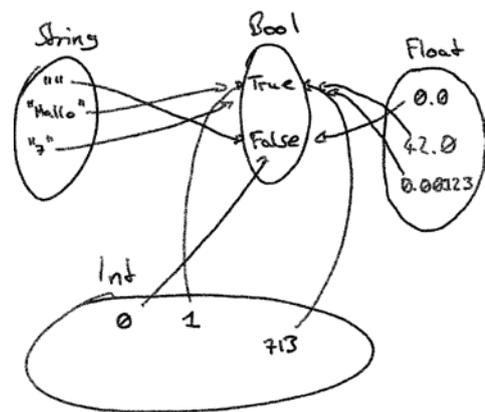
- Fallunterscheidungen mit `if <condition>: <then> else: <else>`
- was eine Condition (Fallunterscheidung) ist
- Das jeweils nur <then> (exklusives) oder <else> ausgeführt wird.

Code

```
if name == "Inspirer":  
    print("Oh, dich habe ich doch schon einmal gesehen!")  
else:  
    print("Hallo, " + name)
```

Datentypen und Funktionen

- Jetzt wird es ein bisschen komplizierter, und die Kids müssen ihre Intuition testen.
- Optional: "Was steht da eigentlich ein Wirklichkeit hinter dem if?" -> lasst sie andere Ausdrücke ausprobieren, die sie aus der Mathematik kennen, und/oder fragt sie, ob sie schon einmal etwas von logischen Operatoren gehört haben.
- Um dann schließlich das Ganze mit **True**, **False** und Datentypen (String, Integer, Float, Boolean) aufzulösen, und ihnen eine Funktionsdarstellung mit Mengendiagrammen aufzumalen.



- (über Bildschirm teilen und Paint/aggie.io/Zoom drawing Tool)
- Optional: Zeigt ihnen gerne auch, wie weit die Umwandlung in Python geht. Auch **1**, **0** und **""** haben Wahrheitswerte (**if 1:**, **if "Hallo":**, **if 0:**)
- Was ist eigentlich eine Funktion? (Sollte aus dem Matheunterricht bekannt sein, also könnt ihr die Kinder auch gerne fragen, was sie sich unter einer Funktion vorstellen) => Ein Ding, das, wenn man es fragt, einem etwas zurückgibt.
Konkrete Repräsentation bei uns: Ein "Stück" Code, der anhand von Parametern einen Wert "berechnet" und zurückgibt.
In unserem Fall: eine Zahl $x * 2$.
Beziehungsweise einen Bool: True oder False.
- Die Funktion malZwei() wird in der Kleingruppe live gecodet.
- Lasst die Kids danach ihre erste eigene Funktion schreiben: **betrag(int zahl)** soll den Betrag der angegebenen Zahl zurückgeben. (optional: signum bzw. istPositiv)

Was sie lernen

- Wie man Funktionen definiert: **def <name>(*params): <body>**
- was ein Datentyp ist; welche Datentypen es u.a. gibt (Int, Float, String, Bool)
- Funktionen sind mehr als nur Graphen in Koordinatensystemen (mehr als das, was sie bisher in Mathe hatten).

Code

```
# Diese Funktion wird live gecoded
def malZwei(x):
    x = int(x)
    return 2*x

# Könnt ihr bei interessierten Gruppen zusätzlich live coden (+erklären)
def to_bool(x):
    if x == "":
        return False
    if x == 0:
```

```

        return False
    if x == None:
        return False
    else:
        return True

# malZwei(2) => 4
# malZwei(4) => 8

# Das sollten die Teilnehmenden am Ende dieser Einheit selber schreiben
def betrag(x):
    if x < 0:
        return -x
    if x >= 0:
        return x

```

Wenn ihr hier bei in etwa der Hälfte der Zeit seid (und noch keine gemacht habt), würde sich hier eine größere Pause von 30 Minuten anbieten. Dann hätte man eine klare Einteilung zwischen fachlichem Input und Projektarbeit.

Input zu CO₂/Klimawandel

- Eine kleine Fragerunde zum Thema Klima/Klimawandel - was wissen die Kids schon, was fehlt ihnen?
 - Was stößt alles CO₂ aus, und wie viel?
 - Was ist ansonsten relevant?
 - Was ist mit den Kühen? -> gerne ein kleines Gespräch darüber beginnen :)
- kurzer Input zum Thema Treibhausgase:
 - Reflexion von Strahlung an den GreenHouse Gases in der Atmosphäre
 - mehr GHG in der Atmosphäre, mehr Reflexion -> die Atmosphäre wird wärmer
- Achtung! -> um klimaneutral zu werden, muss die gesamte Wirtschaft dekarbonisiert werden. Eine individuelle Reduktion des CO₂-Fußabdrucks ist zwar hilfreich, reicht aber keinesfalls aus.

Den Rest des Kurses werden wir mit [100 Punkte hat ein guter Tag](#) arbeiten. Die Idee ist, dass wenn wir als Menschheit klimaneutral leben wollen, wir alle im Schnitt nicht das Budget von 100 Punkten am Tag überschreiten sollten. Schaut euch die Seite gerne im Vorhinein schon einmal an und zeigt sie auch jetzt den Kindern.

Code lesen

- Die erste "große" Funktion, die die Kids lesen und analysieren dürfen.
- Geht sie am besten im Gespräch mit den Kids Stück für Stück durch, lasst hauptsächlich sie reden, diskutieren, ausprobieren, und lenkt sie dann, wenn es stark in die falsche Richtung geht (inhaltlich oder sozial)

Was sie lernen

- Wie man komplexere Funktionen liest und versteht
- Dass Funktionen eigentlich nahezu beliebige, wiederverwendbare Codeblöcke sind, die man bloß nicht so oft tippen möchte.
- Dass Funktionen optimalerweise eine Sinneinheit bilden
- Was Kommentare sind und wie sie funktionieren

Code

```
def CO2_berechnen(mult, weg):
    punkte = mult*weg
    return punkte

def zugfahrt(multiplikator):
    strecke = input("Wie weit fährst du mit dem Zug? (in km)")
    return CO2_berechnen(multiplikator, int(strecke))

punkte_pro_km = 1.4 # Punkte vom CO2 Budget
points = zugfahrt(punkte_pro_km)
print("Durch deine Fahrt mit dem Zug hast du " + str(points) + " Punkte verbraucht.")
```

Projektarbeit

- Von hier an ist alles offen. Die Kids dürfen gerne ihre eigenen Funktionen bauen, zusätzliche Verkehrsmittel hinzufügen, Pflanzen einbauen oder eine längere Fragerunde daraus machen.
- Wahrscheinlich seid ihr aber dann auch hier am Ende angekommen, vor allem, wenn ihr von den Kids genug ausgefragt wurdet.

(Größeres) Beispielprojekt

- Dieses Projekt ist nur ein Beispielprojekt, lasst die Kids gerne ihre eigenen Ideen ausprobieren.
- Dieses Projekt sollte man auch nicht als einen Erwartungshorizont verstehen, wenn die Kids nicht "so weit" kommen, ist das auch vollkommen okay.
- Andere Ideen, über die die Teilnehmenden noch nachdenken könnten, wären:
 - Wie kriege ich verschiedene Transportmittel in mein Programm?
 - Welche Freizeitaktivitäten haben welchen CO₂-Ausstoß?
 - Für Schnelle: CO₂-Budget von einem Urlaub, also Stück für Stück über Transportmittel, Länge der Reise, Aktivitäten, Essen, etc.
 - Wie sieht es mit Pflanzen aus? Gibt es auch einen positiven CO₂ Abdruck?
 - Es gibt hier viel Code, der sich wiederholt. Geht das besser? Wenn ja, wie?

```
def CO2_berechnen(mult, weg):
    punkte = 0
```

```

punkte = mult*weg
return punkte

def zugfahrt(multiplikator):
    strecke = input("Wie weit fährst du mit dem Zug? (in km)")
    return CO2_berechnen(multiplikator, int(strecke))

def autofahrt(multiplikator):
    strecke = input("Wie weit fährst du mit dem Auto? (in km)")
    return CO2_berechnen(multiplikator, int(strecke))

def fahrradfahrt(multiplikator):
    strecke = input("Wie weit fährst du mit dem Fahrrad (in km)?")
    return CO2_berechnen(multiplikator, int(strecke))

vkmittel = input("Mit was fährst du? Auto, Zug oder Rad? ")
if vkmittel == "Zug":
    punkte_pro_km_zug = 1.4 # Punkte vom CO2 Budget
    points = zugfahrt(punkte_pro_km_zug)
    print("Durch deine Fahrt mit dem Zug hast du " + str(points) + " Punkte
    verbraucht.")
elif vkmittel == "Auto":
    punkte_pro_km_auto = 2 # Punkte vom CO2 Budget
    points = autofahrt(punkte_pro_km_auto)
    print("Durch deine Fahrt mit dem Auto hast du " + str(points) + " Punkte
    verbraucht.")
elif vkmittel == "Rad":
    punkte_pro_km_rad = 0 # Punkte vom CO2 Budget
    points = fahrradfahrt(punkte_pro_km_rad)
    print("Durch deine Fahrt mit dem Rad hast du " + str(points) + " Punkte
    verbraucht.")
else:
    print("Dieses Verkehrsmittel kenne ich leider nicht :)")

```

Abschlussrunde (Plenum, ~20 min)

- Die Kids stellen ihre Projekte vor (einer pro Gruppe), und erzählen ein bisschen, was sie so gemacht haben.
- Die Inspirer reden über sich, wie sie zur IT gekommen sind, was ihnen daran Spaß macht, wieso sie es weiterhin machen wollen, und was sie den Jugendlichen gerne mitgeben möchten.
- Wir verteilen den Feedbackbogen-Link der Hacker School an die Teilnehmer*innen und den*die Lehrer*in.

5. Feedback und Verabschiedung

15 Min.

Führen Sie abschließend eine Feedbackrunde mit den Schülerinnen durch:

- Was nehmt ihr mit?
- Was hat euch am meisten Spaß gemacht?
- Welche Erkenntnisse konntet ihr sammeln?
- Wie schätzt ihr eure Fähigkeiten im Vergleich zum Anfang ein?

Die Fragen eignen sich sowohl für den persönlichen Austausch als auch zur Erstellung einer **Wortwolke in Mentimeter**. Wörter in der Wolke werden umso größer, je häufiger sie genannt werden.

Beispiel: [mentimeter.com](https://www.mentimeter.com)



Im Nachgang

Damit die Eindrücke des Kurses möglichst lange erhalten bleiben:

- Verschicken Sie [Teilnahmebestätigungen](#)! Diese können Sie im Materialcenter bestellen und postalisch versenden oder über Ihr persönliches Konto generieren und per E-Mail verschicken.
- Verschicken Sie den Link zum Miro-Board, damit die Schülerinnen sich auch nach dem Kurs noch informieren können. Verschicken Sie außerdem einen Screenshot des padlets mit den Steckbriefen sowie einen Screenshot der Wortwolke, um die Eindrücke nachhaltig zu sichern.
- Bieten Sie die Möglichkeit zur erneuten Kontaktaufnahme im Nachgang – so können Sie individuell auf interessierte Schülerinnen eingehen, die weitere Fragen zu bestimmten Berufen oder zum Unternehmen haben. Regen Sie bei Interesse an, dass die Schülerinnen Kontaktdaten austauschen, um in Kontakt zu bleiben.



Unsere Vision ist es, allen Menschen gleiche Verwirklichungschancen zu ermöglichen – im Beruf und für ihre Lebensplanung. Um dies zu erreichen, schaffen wir Erfahrungsräume für Menschen. Wir setzen uns dafür ein, dass Talente und Interessen der jungen Menschen ihre Berufs- und Lebenswegentscheidung bestimmen – nicht ihr Geschlecht.

Weitere Informationen finden Sie auf girls-day.de



Die Hacker School leistet einen wichtigen Beitrag zur digitalen Bildung unserer Jugend in ganz Deutschland, indem sie junge Menschen für die Welt der Programmiersprachen begeistert. Die Vision der Hacker School: Jeder junge Mensch soll in seiner Bildungslaufbahn mindestens eine Zeile Code geschrieben haben, wissen, wie Computer und Programme aufgebaut sind und ausprobieren, ob IT etwas für die persönliche Zukunft ist.

Weitere Informationen finden Sie auf hacker-school.de

Links zu weiteren Projekten und Informationen rund um das Thema digitale Kompetenzen und Informatik

- [Initiative YouCodeGirls](#)

Die Initiative YouCodeGirls, die im Oktober 2020 gestartet ist, möchte Mädchen und junge Frauen für das Thema Coding begeistern. Ziel ist es, eine digitale Lern- und Lehrplattform zu entwickeln, auf der sie sich unkompliziert, angeleitet und passend zu ihren Bedürfnissen weiterbilden können. Dabei werden bereits bestehende Lernangebote zum Thema Coding gebündelt und durch neu entwickelte, innovative Formate ergänzt. Das Bundesministerium für Familie, Senioren, Frauen und Jugend (BMFSFJ) fördert YouCodeGirls als Leuchtturmprojekt der smarten Gesellschaftspolitik.

- [Femtec. Alumnae e.V. Girls macht MI\(N\)T!](#)

Der Femtec.Alumnae e.V. (kurz: FTA) ist ein Netzwerk hochqualifizierter, technikbegeisterter und engagierter Frauen mit einem akademischen Hintergrund in MINT-Fächern (Mathematik, Informatik, Naturwissenschaft und Technik). Ziel ist, Schülerinnen zu ermutigen, sich für eine Berufslaufbahn im MINT-Bereich zu entscheiden. Hierzu gibt es Angebote zum Experimentieren, u.a. im Bereich Informatik.

- [Roberta Initiative](#)

»Roberta – Lernen mit Robotern« ist eine Bildungsinitiative des Fraunhofer-Instituts für Intelligente Analyse- und Informationssysteme IAIS. Seit 2002 verfolgt das Fraunhofer IAIS mit Roberta das Ziel, mehr Kinder und Jugendliche für Inhalte und Möglichkeiten der Technik und Naturwissenschaften zu begeistern. Indem sich Mädchen und Jungen in gendergerechten Roberta-Kursen mit Robotern oder Mikroprozessoren auseinandersetzen, sie zusammenbauen und programmieren, erhalten sie spielerisch Zugang zu den MINT-Fächern: Mathematik, Informatik, Naturwissenschaften und Technik.

- [Cybermentor](#)

CyberMentor ist Deutschlands größtes Online-Mentoring-Programm für Mädchen in MINT. Bis zu 800 Schülerinnen der 5. bis 13. Klasse werden ein Jahr von einer persönlichen Mentorin begleitet. Sie regt als Rollenvorbild zu MINT-Aktivitäten an und gibt Hinweise zur Studien- und Berufswahl. Das Mentoring erfolgt über eine geschützte Online-Plattform mit Mail, Chat und Forum und wird durch vielfältige Informationen zu MINT, Studium und Berufswahl unterstützt.

- [CODE GIRLS](#)

Die Code Girls organisieren ehrenamtlich Workshops und Vorträge für Einsteigerinnen rund um das Themen Code, Programmierung und Digitalisierung. Sie sind vor allem im Raum Leipzig aktiv, stellen aber auf ihrer Website auch kostenloses Material für Einsteigerinnen zur Verfügung.

- [CODE+DESIGN INITIATIVE E.V.](#)

Die Code+Design Initiative begeistert Jugendliche für das gemeinsame Entdecken der digitalen Welt und begleitet sie auf ihrem Weg in die Jobs von Morgen. Durch bewährte Angebote werden Jugendliche eingeladen, digitale Technologien zu entdecken, sich digitale Grundkompetenzen sowie eine ausgeprägte Medienkompetenz, eine „Digital Literacy“, anzueignen. Sie entwickeln gleichzeitig ein Selbstbewusstsein für die eigenen gestalterischen Fähigkeiten und können sich im MINT-Bereich und der Digitalwirtschaft beruflich orientieren.

- [Komm, mach MINT Informatikbroschüre](#)

Informatikerinnen in unterschiedlichen Karrierephasen und Arbeitsfeldern erzählen, was Informatik eigentlich ist, in welchen Bereichen Informatikerinnen arbeiten und welche Voraussetzungen sie dafür mitbringen mussten. Außerdem berichten Auszubildende und Studentinnen von ihren Erfahrungen im Berufs- und Studienalltag. Hier erfahren Schülerinnen mehr über Medizininformatik, Robotik oder Webentwicklung und entdecken selbst, warum die Informatik für unser alltägliches Leben so wichtig ist.

- [Studie „Digital Gender Gap - Lagebild zu Gender\(un\)gleichheiten in der digitalisierten Welt“](#)

Die Studie "Digital Gender Gap" ist eine aus der Zusammenarbeit der Initiative D21 und des Kompetenzzentrums Technik-Diversity-Chancengleichheit entstandene Sonderauswertung der Studie D21-Digital-Index 2018 / 2019. Die Studie gibt Empfehlungen, wie Akteurinnen und Akteure im Bildungsbereich sowie in Unternehmen und Organisationen ihren Teil zu mehr Chancengleichheit in einer digitalisierten Arbeitswelt beitragen können.

Anhang – Digitale Kompetenzen nach DigComp (EU)

Kompetenzbereiche	Kompetenzen	Beispiele
1. Informations- und Datenkompetenz	1.1. Durchsuchen, Suchen und Filtern von Daten, Informationen und digitalen Inhalten	Ich weiß, wie ich Suchmaschinen nutze, um bestimmte Informationen zu erhalten (z.B. mit bestimmten Suchbegriffen).
	1.2. Auswertung von Daten, Informationen und digitalen Inhalten	Ich kenne Websites und Datenbanken, auf/in denen ich verlässliche Informationen zu verschiedenen Berufen oder Literatur für meine Hausaufgaben finde.
	1.3. Verwaltung von Daten, Informationen und digitalen Inhalten	Ich kann Links zu Websites und Datenbanken so organisieren, dass ich sie jederzeit wiederfinde (z.B. im Browser oder mit Apps).
2. Kommunikation und Zusammenarbeit	2.1. Interaktion durch digitale Technologien	Ich kann Messenger (z.B. Whatsapp, Facebook) dazu nutzen, um Gruppenarbeiten mit anderen Schüler*innen zu organisieren (u.a. Gruppen zusammenstellen).
	2.2. Teilen durch digitale Technologien	Ich weiß, wie ich digitale Inhalte über cloudbasierte Software mit anderen teile.

	2.3. Bürger*innenschaft durch digitale Technologien	Ich kann z.B. Mikroblogs (u.a. #Twitter) dazu nutzen, um mich für soziale Themen stark oder andere auf bestimmte Themen aufmerksam zu machen.
	2.4. Zusammenarbeit durch digitale Technologien	Ich weiß, welche Tools für die Zusammenarbeit in digitaler Umgebung geeignet sind und kann diese nutzen, um gemeinsam mit anderen Schüler*innen an einem Projekt zu arbeiten (z.B. Film erstellen, Aufgaben verteilen).
	2.5. Netiquette	Ich kenne Verhaltensregeln, die für die Zusammenarbeit in digitaler Umgebung wichtig sind (z.B. „Mikrofon-Regelungen“, Respekt im Umgang miteinander, sachliche Kritik, klare Ausdrucksweisen, denn Gestik und Mimik werden in Schriftsprache nicht mittransportiert).
	2.6. Verwaltung der digitalen Identität	Ich weiß, was eine digitale Identität ist (u.a. Inhalte, die einer Person im Internet zugeordnet werden können), und kenne Grundlagen, wie ich diese schützen kann (z.B. sichere Passwörter).
3. Erstellung digitaler Inhalte	3.1. Entwicklung digitaler Inhalte	Ich kann digitale Inhalte in verschiedenen Formaten erstellen (z.B. Präsentation, Instagram-Story oder Video).

	3.2. Integration und Überarbeitung digitaler Inhalte	Ich kann z.B. animierte Präsentationen überarbeiten und neue Inhalte hinzufügen oder die Präsentation in andere digitale Tools einbetten (z.B. Whiteboard).
	3.3. Urheberrecht und Lizenzen	Ich weiß, welche Foto-Datenbanken ich nutzen kann, um lizenz- und kostenfreie Fotos für meine Präsentation herunterzuladen.
	3.4. Programmierung	Ich verstehe die Grundlagen des Programmierens (Programme, Apps werden durch eine Abfolge von Anweisungen „gebaut“. Es gibt dafür verschiedene Programmiersprachen) und kann selbst ein (einfaches) Programm schreiben.
4. Sicherheit	4.1. Geräte schützen	Ich weiß, wie ich meine Geräte und digitalen Inhalte schütze, z.B. mit sicheren Passwörtern, Virenschutz oder entsprechenden Einstellungen der Privatsphäre.
	4.2. Schutz personenbezogener Daten und Privatsphäre	Ich kann unterscheiden zwischen Inhalten, die ich mit anderen teilen kann und solchen, die ich nicht teilen sollte, um mich und andere zu schützen (z.B. Liste mit Adressdatendaten aller Schüler*innen meiner Klasse).

	4.3. Gesundheit und Wohlbefinden schützen	Ich kenne physische und psychische Gesundheitsrisiken, die durch die Nutzung digitaler Technologien entstehen können (ständige Erreichbarkeit, Entgrenzung, Cyber-Mobbing) und kenne einfache Wege, diese möglichst zu vermeiden.
	4.4. Umweltschutz	Mir ist bewusst, welchen Einfluss die Nutzung digitaler Technologien auf die Umwelt hat (z.B. Herstellung von Computern, Smartphones benötigt viel Energie und Ressourcen).
5. Problemlösung	5.1. Technische Probleme lösen	Ich kann (einfache) technische Probleme bei der Nutzung von Geräten/ digitalen Plattformen identifizieren und ggf. selbst lösen. Beispiele: Drucker funktioniert nicht / Upload eines Fotos funktioniert nicht – Datei zu groß?
	5.2. Bedürfnisse und technologische Reaktionen identifizieren	Ich kann die digitale Umgebung (z.B. ein Matheprogramm) meinen Bedürfnissen anpassen, z.B. die Sprache auswählen, die Schrift vergrößern.
	5.3. Kreative Nutzung digitaler Technologien	Ich kann z.B. verschiedene Tools auf Lernplattformen nutzen, um mein Wissen zu vergrößern und Perspektiven zu erweitern (Wiki, Austauschforum, Whiteboard).

	5.4. Identifizieren digitaler Kompetenzlücken	Ich erkenne, was ich noch nicht so gut kann bzw. wo ich Wissenslücken habe und was ich verbessern sollte, z.B. Suche von Daten und Informationen oder Schutz von Geräten und Daten.
--	---	---

Quellen

EU SIENCE HUB – The European Commission’s science and knowledge service: [The Digital Competence Framework 2.0](#)

Stephanie Carretero & Riina Vuorikari & Yves Punie, 2017. "[DigComp 2.1: The Digital Competence Framework for Citizens with eight proficiency levels and examples of use](#)", JRC Working Papers JRC106281, Joint Research Centre.