Coding Lessons For Children

As computer software continues to play an important part in modern life, an understanding of computer programs is essential.

We are pleased to offer computer programming (coding) lessons for children from the age of 8 upwards.

Delivery Mode

- Live instruction via Zoom
- Hands-on learning
- Fun and engaging activities

What Will The Lessons Cover?

The lessons will cover foundational coding concepts such as:

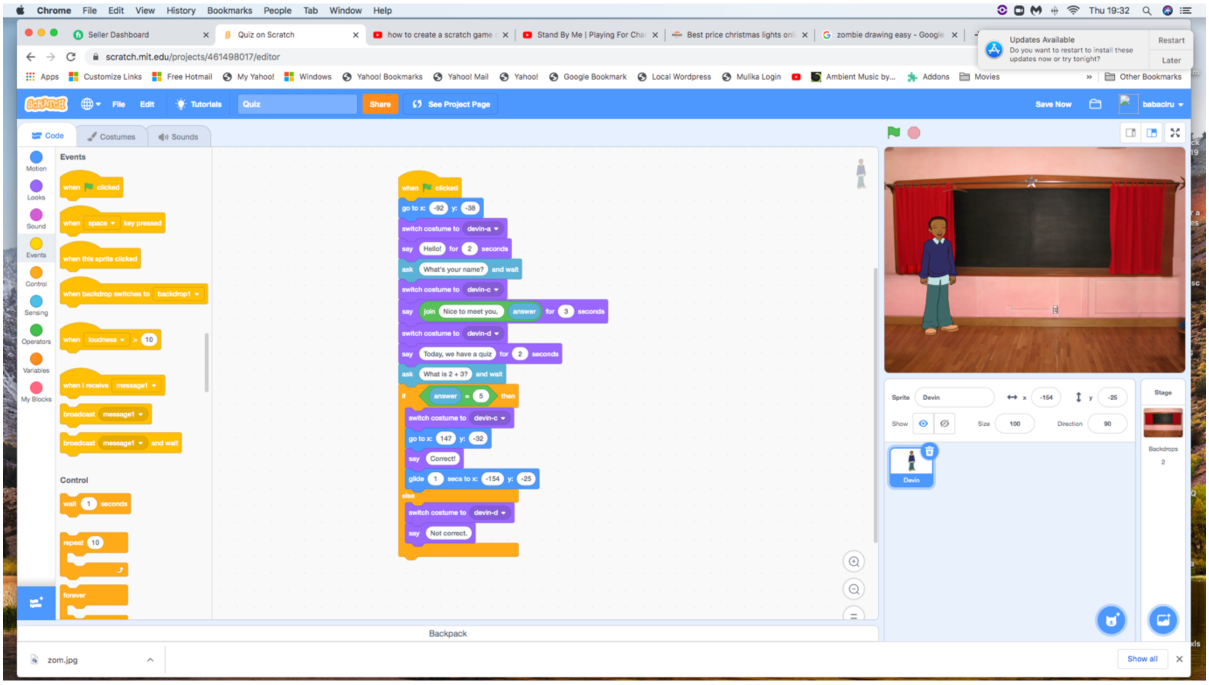
- receiving input
- output
- program flow - sequential execution
- program flow - conditions
- program flow - repetition or loops,
- co-ordinates (if using *Scratch*)
- variables
- operators
- logical steps in coding

If so desired, we can cover specific portions of your child's school computer science syllabus.

The learners will build a number of fun projects that will incorporate these concepts in an engaging manner.

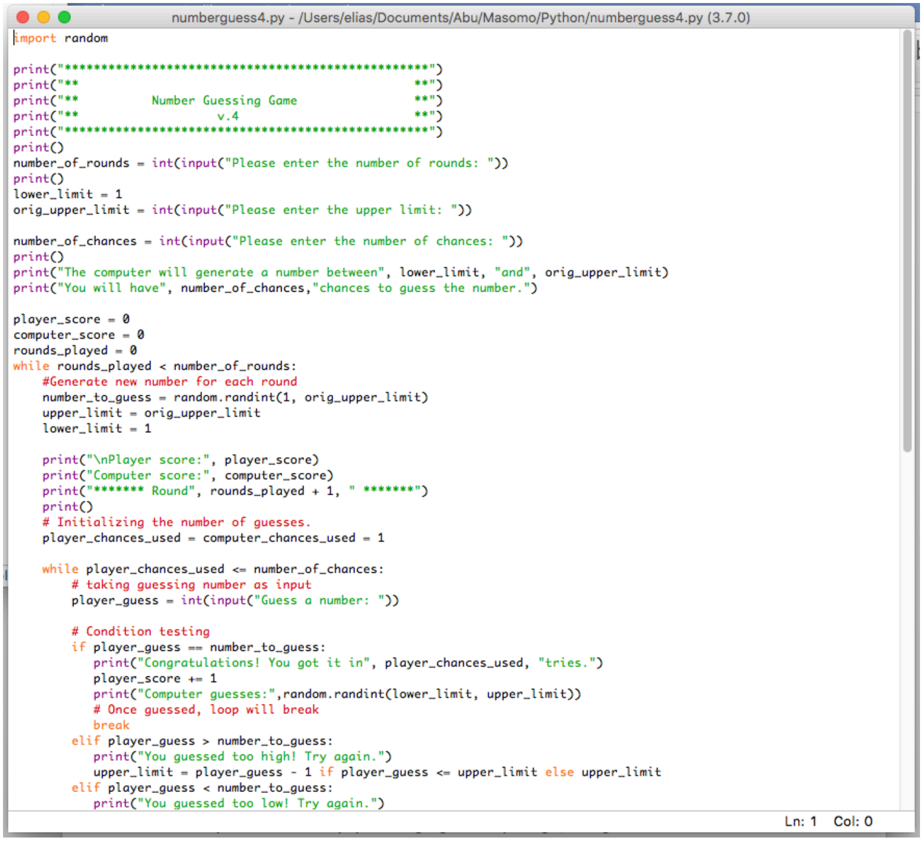
At the end of the course, the children should be able to build a simple program or game and will have developed an understanding of how software works**.**

For children aged 8 to 13, we will use the *Scratch* programming language. *Scratch* is a visual programming language developed by MIT that is designed to teach children coding. It uses blocks of code to build programs.



```
when [flag] clicked
go to x: -92 y: -38
switch costume to devin-a
say Hello! for 2 seconds
ask What's your name? and wait
switch costume to devin-c
say join Nice to meet you, answer for 3 seconds
switch costume to devin-d
say Today, we have a quiz for 2 seconds
ask What is 2 + 3? and wait
if answer = 5 then
    switch costume to devin-c
    go to x: 147 y: -32
    say Correct!
    glide 1 secs to x: -154 y: -25
else
    switch costume to devin-d
    say Not correct.
```

Older children may also use *Scratch*, or may choose to learn *Python* instead. *Python* is currently one of the IT industry’s most popular programming languages use by companies such as Google and Instagram, for things ranging from web development, automation, data science to Artificial Intelligence.



numberguess4.py - /Users/elias/Documents/Abu/Masomo/Python/numberguess4.py (3.7.0)

```
import random

print("**************************************************")
print("**                                              **")
print("**          Number Guessing Game                **")
print("**                   v.4                        **")
print("**************************************************")
print()
number_of_rounds = int(input("Please enter the number of rounds: "))
print()
lower_limit = 1
orig_upper_limit = int(input("Please enter the upper limit: "))

number_of_chances = int(input("Please enter the number of chances: "))
print()
print("The computer will generate a number between", lower_limit, "and", orig_upper_limit)
print("You will have", number_of_chances,"chances to guess the number.")

player_score = 0
computer_score = 0
rounds_played = 0
while rounds_played < number_of_rounds:
    #Generate new number for each round
    number_to_guess = random.randint(1, orig_upper_limit)
    upper_limit = orig_upper_limit
    lower_limit = 1

    print("\nPlayer score:", player_score)
    print("Computer score:", computer_score)
    print("******* Round", rounds_played + 1, " *******")
    print()
    # Initializing the number of guesses.
    player_chances_used = computer_chances_used = 1

    while player_chances_used <= number_of_chances:
        # taking guessing number as input
        player_guess = int(input("Guess a number: "))

        # Condition testing
        if player_guess == number_to_guess:
            print("Congratulations! You got it in", player_chances_used, "tries.")
            player_score += 1
            print("Computer guesses:",random.randint(lower_limit, upper_limit))
            # Once guessed, loop will break
            break
        elif player_guess > number_to_guess:
            print("You guessed too high! Try again.")
            upper_limit = player_guess - 1 if player_guess <= upper_limit else upper_limit
        elif player_guess < number_to_guess:
            print("You guessed too low! Try again.")
```

Ln: 1   Col: 0

Whichever the language used, the children will learn and develop skills such as

- Coding

- Analytical thinking
- Concentration
- Persistence
- Problem-solving
- Attention to detail.

The children will develop an understanding of how the software that drives modern devices works.

Let your children spend their time in a fun and educational activity.

Please sign up at the link below
www.codingclubke.com or email info@codingclubke.com