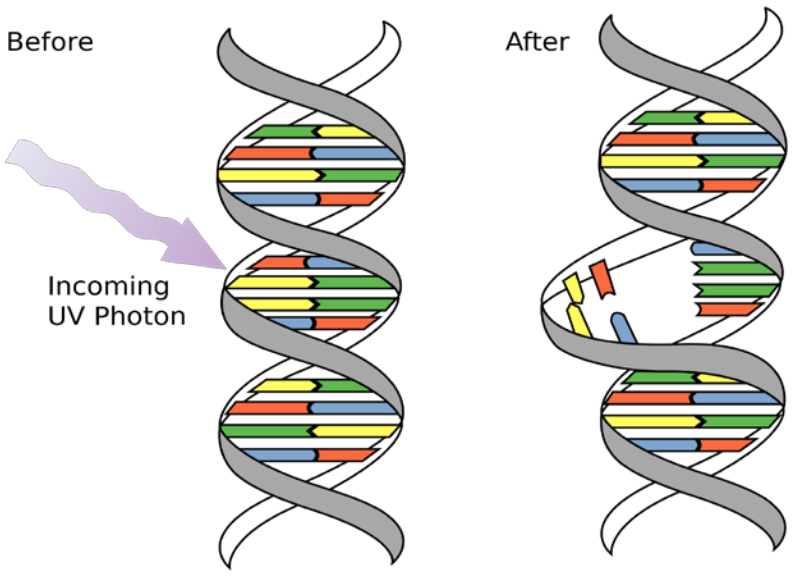
Python Coding Lesson

Lesson Title:	**DNA Mutations**
Grade Level:	**High School**
Subject:	**Biology**
Contributed By:	**C. Welborn and J. Muntazar**

Engagement:

- Students will be shown clips of X-Men - to peak their interest and show them DNA modification.
    - Ask students what comes to mind when they hear the word “mutation” or “mutant”
    - Ask students how the characters in the movie clip became a mutant
        - I.e. where in their bodies may a change happen
- A detailed explanation of each term when it arises during the lesson will be given.
- Inform students of the learning targets for this lesson:
    - identify mutagenic factors that can alter DNA.
    - distinguish between different types of mutations.
- How does DNA sequencing affect mutations?



Source: https://knowgenetics.org/dna-mutations-2/

This material is based upon work supported by the Office of Naval Support under Award No. N00014-17-1-2948.

Python Coding Lesson

Exploration:

- To better understand DNA mutations, students will be able to add into a coding skeleton and create a program using functions in which they can replicate the different types commonly found DNA mutations.
    - Base substitutions
    - Deletions
    - insertions
- After a function is created, students should run their program to validate their mutation.
    - There is a sequence pattern in which students will validate to find out what type of DNA mutation they are using within the program.
    - Base sequence should show that “A” “T” “C” and “G” should be the only four consistent letters present.
- Students should be able to use their program as an artificial tool to mimic one of the three common DNA mutations.
    - The coding program will recognize four base pair letters ‘A’ ‘T’ ‘C’ and ‘G’ only. Because there are only four that are known to match up with their specific letter partner, students can show they figured out which mutation they are representing when the program runs.
- Describe the student-led, hands-on/minds-on activities students will be doing.

Explanation:

- Bringing the class back together, students will share their coding mutation that they have replicated.
- As students run their mutation code, the teacher will remind them to implement the proper vocab required for the lesson during class discussion.
    - The more vocab terms used in complete sentences will help students become better familiarized with the terms...
- During this, the following questions that will help clarify some misunderstandings will be asked:
    - What is an insertion mutation (point, deletion, etc.)?
    - Which of these examples accurately shows an insertion mutation?
    - Which, if any, of these examples misrepresent an insertion mutation?
    - What does a mutation change?
    - What does a change to DNA do? (Think back to Translation)

This material is based upon work supported by the Office of Naval Support under Award No. N00014-17-1-2948.

- Students share out their understanding and findings before any terms or explanations are given.

Elaboration:

- Using their knowledge about DNA mutations and observing their code running to mimic a mutation, teacher will ask a few more questions to encourage a deeper connection between DNA mutations and their effects on humans:
    - What are the effects of mutations in an individual?
    - Is it possible for mutations to have be beneficial?
    - Can mutations be passed to offspring?
    - What are the long-term/multi-generational effects of a mutation?

Evaluation:

- Exit ticket 1: Conceptual- Why might a mutation of a sperm cell or egg cell have greater consequences than a mutation of a skin or lung cell?
- Exit ticket 2: similar to question below where students will be able to use Python to build a base pair:
    - Which of the scenarios represented below represent an inversion mutation?
        - From the given sequence of DNA, “ATT CGG ATC”, create: a. base pair addition
        - b. base pair deletion
        - c. base pair substitution

This material is based upon work supported by the Office of Naval Support under Award No. N00014-17-1-2948.