## Verzweigungen mit Turtle - unter Verwendung von Zufallszahlen

### Aufgabe 1 - Anderes Verhalten im Wasser als am Sand (if mit else)

Nachdem unsere Turtle sich im Sand abgekühlt ist sie hungrig. Sie sucht nach Blumen eines Kaktus in der Wüste und nach im Meer nach Algen und Seegras. Sie ist wie zuvor im Haus am Rande des Meers. Links von Ihr ist das Meer und rechts von ihr die Wüste. \* Da sie wieder nicht weiß wohin, versucht sie sich wieder mit zufälligen Positionen. \* Egal ob Wasser oder Wüste, sie legt wieder einen Faden ab und befestigt diesen im Garten indem sie sich in den Boden drückt. Sie hat dabei die bereits verwendete Form der Turtle. \* Da sie hungrig ist, geht sie am Sand mit der doppelten Geschwindigkeit welche am Land üblich ist. Wenn die Turtle ins Meer geht, dann schwimmt sie und ist um das 50-fache Schneller als die normale Geschwindigkeit am Land. Im Meer ist es egal ob sie hungrig ist oder nicht, da sie immer so schnell ist. \* Wenn die Turtle im Meer schwimmt, dann sehen wir keine Beine und deshalb nur einen Kreis. Ansonsten stelle sie als Pfeil dar, denn Sie ist doppelt so schnell als sonst. \* Am schluss taucht sie im Wasser, am Land jedoch bleibt sie einfach stehen.

**Hinweise:** Verwende Variablen für:

* die Geschwindigkeit der Turtle am Wasser, am Land und hungrig am Land,
* die Form im Garten, in der Wüste und im Wasser.

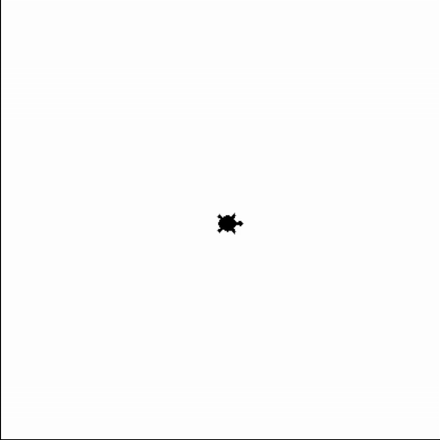
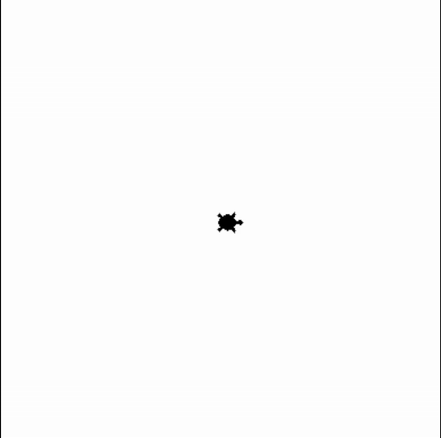
Entscheide selbst ob hier Variablen für \* den logischen Ausdruck der Bedingung unserer Verzweigung verwendet werden sollen.

Verwende dazu folgende Prozeduren:

* pendown(): Die Turtle legt **einen** Faden am Boden ab. Diese malt dadruch Linien wenn diese sich **später** bewegt.
* goto(-100, 200): Die Turtle bewegt sich in einer *geraden Linie* zu der angegebenen *Position*. Die *Position* wird in *x* und *y* *Koordinaten* abegeben. Hier ist die Mitte des Fensters *x = 0* und *y = 0* ist.
* hideturtle(): Die Turtle gräbt sich ein und versteckt sich.
* stamp(): Die Turtle drückt sich auf den Boden und hinterlässt einen Abdruck.
* shape("turtle") oder shape("circle"): Wir lassen unsere Turtle anders aussehen.

Sowie folgende Funktionen:

* randint(3, 8): Wir ziehen eine zufällige Zahl ohne Kommastellen von z.B. 3 bis 8.
* window\_width(): Gibt die breite unseres Fenstes zurück. Merke dir die Antwort mit einer Variable. Das geht so *breite = window\_width()*.
* window\_height(): Gibt die breite unseres Fenstes zurück. Merke dir die Antwort mit einer Variable. Das geht so *hoehe = window\_height()*.

**Lösung:**

from turtle import \*  
from random import randint  
  
# --- Vorbereitung ---  
# Die Variablen für die Geschwindigkeit.  
geschwindigkeit\_am\_land = 1  
geschwindigkeit\_am\_land\_hungrig = 2 \* geschwindigkeit\_am\_land  
geschwindigkeit\_im\_wasser = 50 \* geschwindigkeit\_am\_land  
  
# Die Variablen für die Form.  
form\_am\_land = "turtle"  
form\_am\_land\_hungrig = "arrow"  
form\_im\_wasser = "circle"  
  
shape(form\_am\_land)   
speed(geschwindigkeit\_am\_land)  
  
# --- Logik ---  
# Wir generierem zufällige Koordinaten innerhalb der Fenstergröße.  
# Das Wort Zufall wird im Englischen das Wort random.   
breite = window\_width()  
hoehe = window\_height()  
  
# Die Bildschrimbreite geht von z.B. 0 bis 100. Für unsre Turtle ist jedoch die Mitte dieser Breite der 0-Punkt.  
# Dadurch ändert sich 0 bis 100 zu -50 bis 50.  
# Um die Turtle zufällig zu navigieren, müssen wir eine Zahl zwischen -50 und 50 ziehen.   
# Das gleiche gilt für die Hoehe.  
  
# Die Ganzzahldivision.  
# Diese ist unter 5 dividiert durch 2 ist 2, mit 1 Rest bekannt und wird mit Python mit // geschrieben.  
# Das 1 Rest wird hier mit // ignorieret.  
halbe\_breite\_ohne\_komma = breite // 2   
halbe\_hoehe\_ohne\_komma = hoehe // 2  
  
# Da wir zufällige Zahlen ohne Komma wollen, verwenden wir die Funktion randint.  
# Diese benötigt zwei zahlen, zwischen denen sie zufällig eine Wählt.   
# randint(3, 8) gibt eine Zahl welche 3 sein kann, 8 sein kann und alles dazwischen. Alles dazwischen ist 4, 5, 6 und 7.  
# Der Name ist eine Kombination aus Random und Integer, was Zufall und Zahl ohne Komma bedeutet.  
ziel\_in\_x = randint(-halbe\_breite\_ohne\_komma, halbe\_breite\_ohne\_komma)  
ziel\_in\_y = randint(-halbe\_hoehe\_ohne\_komma, halbe\_hoehe\_ohne\_komma)  
  
# Die Turtle nimmt den Faden aus der Tasche und legt ihn am Boden wenn sie losgeht.  
pendown()  
  
# Die Turtle drückt den Faden in den boden, fixiert diesen und macht einen Abdruck von sich selbst.  
stamp()  
  
# Wir bewegen die Turtle und passen an mit einer Verzweigung an wie diese dargestellt wird (Kreis oder Turtle) und wie schnell diese ist.   
if ziel\_in\_x > 0:  
 # Setze die Geschwindigkeit auf die Landgeschwindigkeit.   
 speed(geschwindigkeit\_am\_land\_hungrig)  
  
 # Die Turtle geht und ist als Arrow sichtbar.  
 shape(form\_am\_land\_hungrig)   
else:  
 # Setze die Geschwindigkeit auf die Wassergeschwindigkeit.  
 speed(geschwindigkeit\_im\_wasser)  
  
 # Die Turtle schwimmt und es ist nur der Panzer sichtbar. Quasi nur ein Kreis.  
 # Das englische Wort für Kreis ist circle.  
 shape(form\_im\_wasser)   
  
# Wir bewegen uns zur zufällig gewählten Position.  
goto(ziel\_in\_x, ziel\_in\_y)  
  
# Am schluss gräbt sich die Turtle in den Sand oder taucht im Wasser unter.  
hideturtle()  
  
# --- Abschluss ---  
# Schließt das Fenster nicht, wenn das Programm beendet ist.  
done()

### Aufgabe 2 - Anderes Verhalten im Wasser als am Sand und im Norden und Süden (if mit elif und else)

Am nächsten Tag ist die Turtle wieder hungrig und verhält sich auch im prinzip gleich. Jedoch hat die beim vorherigen Tag gemerkt, dass für den sie zu schnell ist. Dort sind scharfe Steine und bilden ein Riff. Sie muss hier genau so langsam wie in der Wüste nach essen suchen. Auch ist es so grell am Land im Süden, denn die Turtle geht genau auf den Sonnenaufgang zu. Sie dreht sich deshalb nach Westen und geht quasi rückwärts an ihr Ziel. Dadurch ist sie genau so schnell wie wenn sie sich ohne Hunger fortbewegen würde.

**Hinweise:** Verwende Variablen für:

* die Geschwindigkeit der Turtle am Wasser, am Land und hungrig am Land,
* die Form im Garten, in der Wüste und im Wasser und
* die logischen Ausdrücke der Bedingungen unserer Mehrfachverzweigung.

Verwende dazu folgende Prozeduren:

* pendown(): Die Turtle legt **einen** Faden am Boden ab. Diese malt dadruch Linien wenn diese sich **später** bewegt.
* goto(-100, 200): Die Turtle bewegt sich in einer *geraden Linie* zu der angegebenen *Position*. Die *Position* wird in *x* und *y* *Koordinaten* abegeben. Hier ist die Mitte des Fensters *x = 0* und *y = 0* ist.
* hideturtle(): Die Turtle gräbt sich ein und versteckt sich.
* stamp(): Die Turtle drückt sich auf den Boden und hinterlässt einen Abdruck.
* shape("turtle"), shape("circle") oder shape("arrow"): Wir lassen unsere Turtle anders aussehen.

Sowie folgende Funktionen:

* randint(3, 8): Wir ziehen eine zufällige Zahl ohne Kommastellen von z.B. 3 bis 8.
* window\_width(): Gibt die breite unseres Fenstes zurück. Merke dir die Antwort mit einer Variable. Das geht so *breite = window\_width()*.
* window\_height(): Gibt die breite unseres Fenstes zurück. Merke dir die Antwort mit einer Variable. Das geht so *hoehe = window\_height()*.

**Lösung:**

from turtle import \*  
from random import randint  
  
# --- Vorbereitung ---  
# Die Variablen für die Geschwindigkeit.  
geschwindigkeit\_am\_land = 1  
geschwindigkeit\_am\_land\_hungrig = 2 \* geschwindigkeit\_am\_land  
geschwindigkeit\_im\_wasser = 50 \* geschwindigkeit\_am\_land  
  
# Die Variablen für die Form.  
form\_am\_land = "turtle"  
form\_am\_land\_hungrig = "arrow"  
form\_im\_wasser = "circle"  
  
shape(form\_am\_land)   
speed(geschwindigkeit\_am\_land)  
  
# --- Logik ---  
# Wir generierem zufällige Koordinaten innerhalb der Fenstergröße.  
# Das Wort Zufall wird im Englischen das Wort random.   
breite = window\_width()  
hoehe = window\_height()  
  
# Die Bildschrimbreite geht von z.B. 0 bis 100. Für unsre Turtle ist jedoch die Mitte dieser Breite der 0-Punkt.  
# Dadurch ändert sich 0 bis 100 zu -50 bis 50.  
# Um die Turtle zufällig zu navigieren, müssen wir eine Zahl zwischen -50 und 50 ziehen.   
# Das gleiche gilt für die Hoehe.  
  
# Die Ganzzahldivision.  
# Diese ist unter 5 dividiert durch 2 ist 2, mit 1 Rest bekannt und wird mit Python mit // geschrieben.  
# Das 1 Rest wird hier mit // ignorieret.  
halbe\_breite\_ohne\_komma = breite // 2   
halbe\_hoehe\_ohne\_komma = hoehe // 2  
  
# Da wir zufällige Zahlen ohne Komma wollen, verwenden wir die Funktion randint.  
# Diese benötigt zwei zahlen, zwischen denen sie zufällig eine Wählt.   
# randint(3, 8) gibt eine Zahl welche 3 sein kann, 8 sein kann und alles dazwischen. Alles dazwischen ist 4, 5, 6 und 7.  
# Der Name ist eine Kombination aus Random und Integer, was Zufall und Zahl ohne Komma bedeutet.  
ziel\_in\_x = randint(-halbe\_breite\_ohne\_komma, halbe\_breite\_ohne\_komma)  
ziel\_in\_y = randint(-halbe\_hoehe\_ohne\_komma, halbe\_hoehe\_ohne\_komma)  
  
# Die Turtle nimmt den Faden aus der Tasche und legt ihn am Boden wenn sie losgeht.  
pendown()  
  
# Die Turtle drückt den Faden in den boden, fixiert diesen und macht einen Abdruck von sich selbst.  
stamp()  
  
# Wir bewegen die Turtle und passen an mit einer Mehrfachverzweigung an wie diese dargestellt wird (Kreis oder Turtle) und wie schnell diese ist.   
im\_wasser = ziel\_in\_x < 0  
am\_land = ziel\_in\_x >= 0  
  
norden = ziel\_in\_y > 0  
sueden = ziel\_in\_y <= 0  
  
schnell\_im\_wasser = im\_wasser and sueden  
langsam\_im\_wasser = im\_wasser and norden  
  
bewegt\_sich\_rueckwaerts\_am\_land = am\_land and sueden  
bewegt\_sich\_vorwärts\_am\_land = am\_land and norden  
  
if schnell\_im\_wasser:  
 # Setze die Geschwindigkeit auf die Wassergeschwindigkeit.   
 speed(geschwindigkeit\_im\_wasser)  
  
 # Die Turtle schwimmt und es ist nur der Panzer sichtbar. Quasi nur ein Kreis.  
 # Das englische Wort für Kreis ist circle.  
 shape(form\_im\_wasser)   
  
 # Wir bewegen uns zur zufällig gewählten Position.  
 goto(ziel\_in\_x, ziel\_in\_y)  
  
 # Am schluss taucht die Turtle im Wasser unter.  
 hideturtle()  
  
elif langsam\_im\_wasser:  
 # Setze die Geschwindigkeit auf die hungrige Landgeschwindigkeit.  
 speed(geschwindigkeit\_am\_land\_hungrig)  
  
 # Die Turtle schwimmt und es ist nur der Panzer sichtbar. Quasi nur ein Kreis.  
 # Das englische Wort für Kreis ist circle.  
 shape(form\_im\_wasser)   
  
 # Wir bewegen uns zur zufällig gewählten Position.  
 goto(ziel\_in\_x, ziel\_in\_y)  
  
 # Am schluss taucht die Turtle im Wasser unter.  
 hideturtle()  
  
elif bewegt\_sich\_rueckwaerts\_am\_land:  
 # Setze die Geschwindigkeit auf die hungrige Landgeschwindigkeit.  
 speed(geschwindigkeit\_am\_land)  
  
 # Die Turtle geht und ist als Arrow sichtbar.  
 shape(form\_am\_land\_hungrig)   
   
 # Die Turtle dreht sich nach Westen um nicht geblendet zu werden.  
 left(180)  
  
 # Wir bewegen uns zur zufällig gewählten Position.  
 goto(ziel\_in\_x, ziel\_in\_y)  
  
elif bewegt\_sich\_vorwärts\_am\_land:  
 # Setze die Geschwindigkeit auf die hungrige Landgeschwindigkeit.  
 speed(geschwindigkeit\_am\_land\_hungrig)  
  
 # Die Turtle geht und ist als Arrow sichtbar.  
 shape(form\_am\_land\_hungrig)   
  
 # Wir bewegen uns zur zufällig gewählten Position.  
 goto(ziel\_in\_x, ziel\_in\_y)  
  
else:  
 # Hier sollten wir nie herkommen! Jedoch kann es passieren, wenn wir Fehler machen.  
 # Wir geben deshalb aus, dass wir hier nie herkommen sollten  
 fehler = "DAS SOLLTE NIE PASSIEREN!"  
 write(fehler)  
 print(fehler)  
  
# --- Abschluss ---  
# Schließt das Fenster nicht, wenn das Programm beendet ist.  
done()

### Sehr Fortgeschritten und sehr Optional - Aufgabe 2.1 - Anderes Verhalten im Wasser als am Sand und im Norden und Süden (verschachteltes if und else)

Weierlesen auf eigene Gefahr. Hier ist es das Ziel den Code ein wenig zu verstehen und nicht selbst zu implementieren. Wir bemerken damit, dass verschachtelte Verzweigungen, sehr schnell, sehr komplex werden können. Es gibt eigene Techniken um hier einen Überblick zu behalten, diese sind jedoch bei weitem nicht sinnvoll hier zu betrachten (Guard Clauses, early Exit und die Verbindung zu De’Morgans Law).

Wir wiederholen die [Aufgabe 2](#Xaca4b1f11dbee904230b6a18c324a49dcb6c000), nur haben wir dort bemerkt, dass in der Mehrfachverzweigung doppelter Code vorkommt. Wir setzen z.B. in jedem Zweig goto(ziel\_in\_x, ziel\_in\_y). Wir können goto(ziel\_in\_x, ziel\_in\_y) aber nicht nach der Mehrfachverzweigung **einmal** aufrufen, denn wenn wir im Wasser sind, dann rufen wir **nach** goto(ziel\_in\_x, ziel\_in\_y) die Funktion hide() auf. Wenn wir es trotzdem tun würden, wäre die Folge, dass wir uns verstecken bevor wir losgehen. Das klingt nicht sinnvoll. Wir können hier jedoch probieren eine Verschachtelung von Verzweigungen zu verwenden. Wie schaffen es dadurch keinen doppelten Code zu erzeugen, jedoch ist der Preis, dass es alles schwerer nachzuvolliehen wird.

**Lösung:**

from turtle import \*  
from random import randint  
  
# --- Vorbereitung ---  
# Die Variablen für die Geschwindigkeit.  
geschwindigkeit\_am\_land = 1  
geschwindigkeit\_am\_land\_hungrig = 2 \* geschwindigkeit\_am\_land  
geschwindigkeit\_im\_wasser = 50 \* geschwindigkeit\_am\_land  
  
# Die Variablen für die Form.  
form\_am\_land = "turtle"  
form\_am\_land\_hungrig = "arrow"  
form\_im\_wasser = "circle"  
  
shape(form\_am\_land)   
speed(geschwindigkeit\_am\_land)  
  
# --- Logik ---  
# Wir generierem zufällige Koordinaten innerhalb der Fenstergröße.  
# Das Wort Zufall wird im Englischen das Wort random.   
breite = window\_width()  
hoehe = window\_height()  
  
# Die Bildschrimbreite geht von z.B. 0 bis 100. Für unsre Turtle ist jedoch die Mitte dieser Breite der 0-Punkt.  
# Dadurch ändert sich 0 bis 100 zu -50 bis 50.  
# Um die Turtle zufällig zu navigieren, müssen wir eine Zahl zwischen -50 und 50 ziehen.   
# Das gleiche gilt für die Hoehe.  
  
# Die Ganzzahldivision.  
# Diese ist unter 5 dividiert durch 2 ist 2, mit 1 Rest bekannt und wird mit Python mit // geschrieben.  
# Das 1 Rest wird hier mit // ignorieret.  
halbe\_breite\_ohne\_komma = breite // 2   
halbe\_hoehe\_ohne\_komma = hoehe // 2  
  
# Da wir zufällige Zahlen ohne Komma wollen, verwenden wir die Funktion randint.  
# Diese benötigt zwei zahlen, zwischen denen sie zufällig eine Wählt.   
# randint(3, 8) gibt eine Zahl welche 3 sein kann, 8 sein kann und alles dazwischen. Alles dazwischen ist 4, 5, 6 und 7.  
# Der Name ist eine Kombination aus Random und Integer, was Zufall und Zahl ohne Komma bedeutet.  
ziel\_in\_x = randint(-halbe\_breite\_ohne\_komma, halbe\_breite\_ohne\_komma)  
ziel\_in\_y = randint(-halbe\_hoehe\_ohne\_komma, halbe\_hoehe\_ohne\_komma)  
  
# Die Turtle nimmt den Faden aus der Tasche und legt ihn am Boden wenn sie losgeht.  
pendown()  
  
# Die Turtle drückt den Faden in den boden, fixiert diesen und macht einen Abdruck von sich selbst.  
stamp()  
  
# Wir bewegen die Turtle und passen an mit einer Mehrfachverzweigung an wie diese dargestellt wird (Kreis oder Turtle) und wie schnell diese ist.   
im\_wasser = ziel\_in\_x < 0  
am\_land = ziel\_in\_x >= 0  
  
norden = ziel\_in\_y > 0  
sueden = ziel\_in\_y <= 0  
  
schnell\_im\_wasser = im\_wasser and sueden  
langsam\_im\_wasser = im\_wasser and norden  
  
bewegt\_sich\_rueckwaerts\_am\_land = am\_land and sueden  
bewegt\_sich\_vorwärts\_am\_land = am\_land and norden  
  
# Alles was für sueden und norden doppelt steht, schreiben wir auf die 1. Ebene am\_land.  
if am\_land:  
 # Die Turtle geht und ist als Arrow sichtbar.  
 shape(form\_am\_land\_hungrig)   
  
 # Es gilt in der 2. Ebene am\_land and sueden, also bewegt\_sich\_rueckwaerts\_am\_land.  
 if sueden:  
 # Setze die Geschwindigkeit auf die hungrige Landgeschwindigkeit.  
 speed(geschwindigkeit\_am\_land)  
  
 # Die Turtle dreht sich nach Westen um nicht geblendet zu werden.  
 left(180)  
  
 # Es gilt in der 2. am\_land and norden, also bewegt\_sich\_vorwärts\_am\_land.  
 if norden:  
 # Setze die Geschwindigkeit auf die hungrige Landgeschwindigkeit.  
 speed(geschwindigkeit\_am\_land\_hungrig)  
  
# Alles was für sueden und norden doppelt steht, schreiben wir auf die 1. Ebene im\_wasser.  
elif im\_wasser:  
 # Die Turtle schwimmt und es ist nur der Panzer sichtbar. Quasi nur ein Kreis.  
 # Das englische Wort für Kreis ist circle.  
 shape(form\_im\_wasser)   
  
 # Es gilt in der 2. im\_wasser and sueden, also schnell\_im\_wasser.  
 if sueden:   
 # Setze die Geschwindigkeit auf die Wassergeschwindigkeit.   
 speed(geschwindigkeit\_im\_wasser)  
  
 # Es gilt in der 2. Ebene im\_wasser and norden, also langsam\_im\_wasser.  
 if norden:  
 # Setze die Geschwindigkeit auf die hungrige Landgeschwindigkeit.  
 speed(geschwindigkeit\_am\_land\_hungrig)  
  
# Wir können nun auch nach der Verzweigung auf die 0. Ebene zurück gehen und nur ein mal goto(ziel\_in\_x, ziel\_in\_y) aufrufen  
# Wir bewegen uns zur zufällig gewählten Position.  
goto(ziel\_in\_x, ziel\_in\_y)  
  
# Wir fragen nochmals ab ob wir im Wasser sind und verstecken die Turtle.  
if im\_wasser:  
 # Am schluss taucht die Turtle im Wasser unter.  
 hideturtle()  
  
# --- Abschluss ---  
# Schließt das Fenster nicht, wenn das Programm beendet ist.  
done()