BIOS1100 H17 uke 6

Lex Nederbragt

Ukens forelesning

- noen praktiske ting
- nytt stoff denne uken
- utvalgte øvelser

Obligatoriske innleveringer

De viktigste endringer:

- du må ikke lenger bestå alle obliger for å kunne gå opp til eksamen, men bare 7 av 11
- vi går bort fra anonyme retting, men fortsetter med tilfeldig fordeling av oppgaver over gruppelærere

Obligatoriske innleveringer

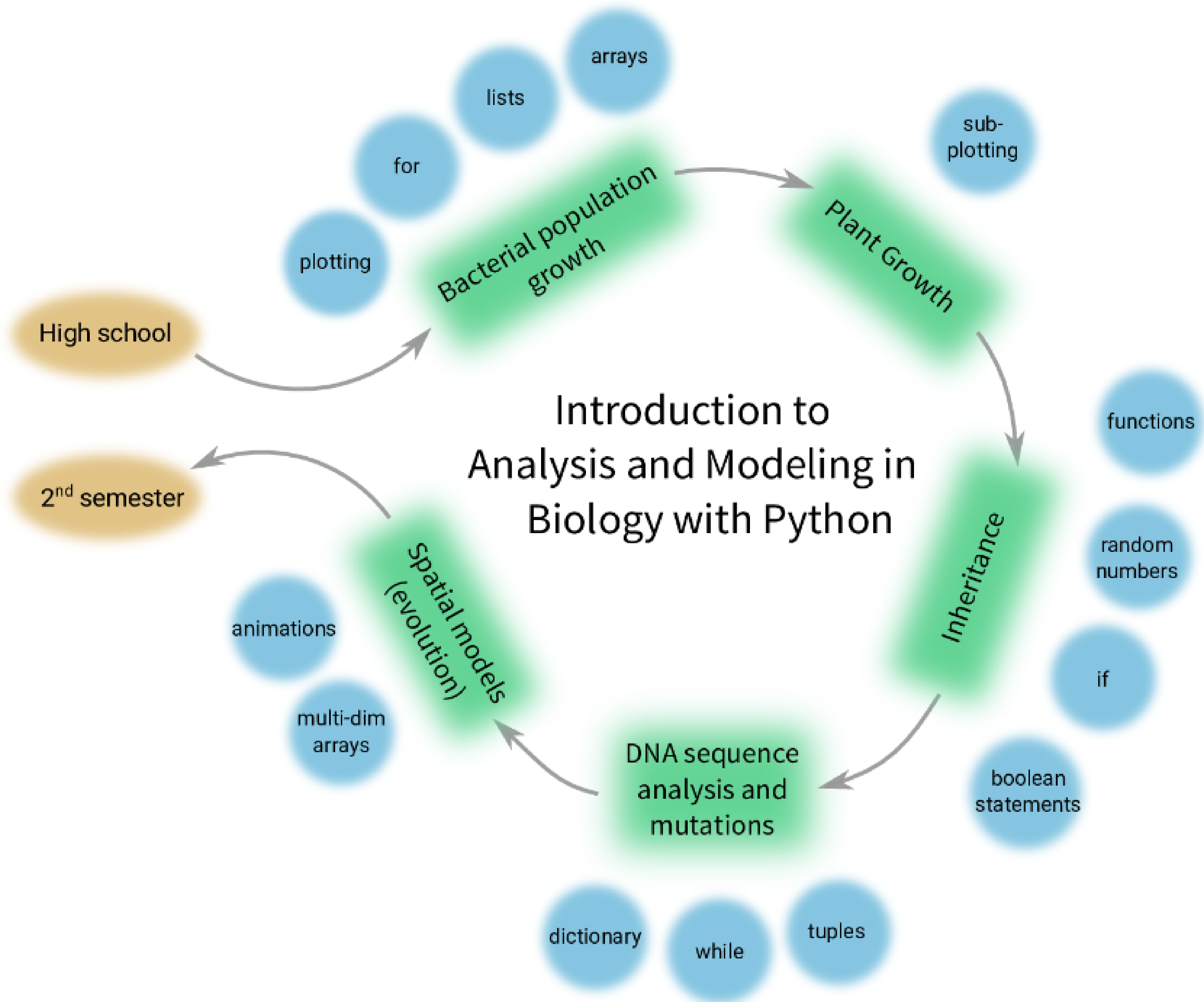
- det er 11 obligatoriske innleveringer i BIOS1100
- obliger er bestått/ikke bestått
- du må bestå 7 av de 11 obligene for å kunne gå opp til eksamen
- innleveringsfrist er tirsdag kl. 23:59
- du skal få svar (bestått/ikke bestått) senest torsdag kl. 23:59
- hvis du får ikke bestått på en oblig får du opp til to sjanser for å levere på nytt
- aller siste leveringsfrist for obliger er tirsdag 23. november kl. 23:59

Obligatoriske innleveringer

De to siste obliger er litt spesielle:

- kursuke 11 (uke 44): oblig 10 med frist 7. november har maks to innleveringer
- kursuke 12 (uke 45): oblig 11 med frist 14. november får du bestått uansett hva du leverer (men du må levere noe, eventuelt en tom notebook)

Undervisningsplan



Læringsmål denne uke

Biologi

- vekst av to-årlige planter

Matematikk

- Kunne lage og implementere andre ordens differenslikninger

Programmering

- kunne lage subplots
- if-tester

Årlig model

- S is the number of seeds per plant
- w is the survival rate
- g is the germination rate
- P_n is the number of plants in year n

$P_n = gwSP_{n-1}$

$P_n = \Delta P \times P_{n-1}$, with $\Delta P = gwS$

Årlig model

$P_n = gwSP_{n-1}$

$P_n = \Delta P \times P_{n-1}$, with $\Delta P = gwS$

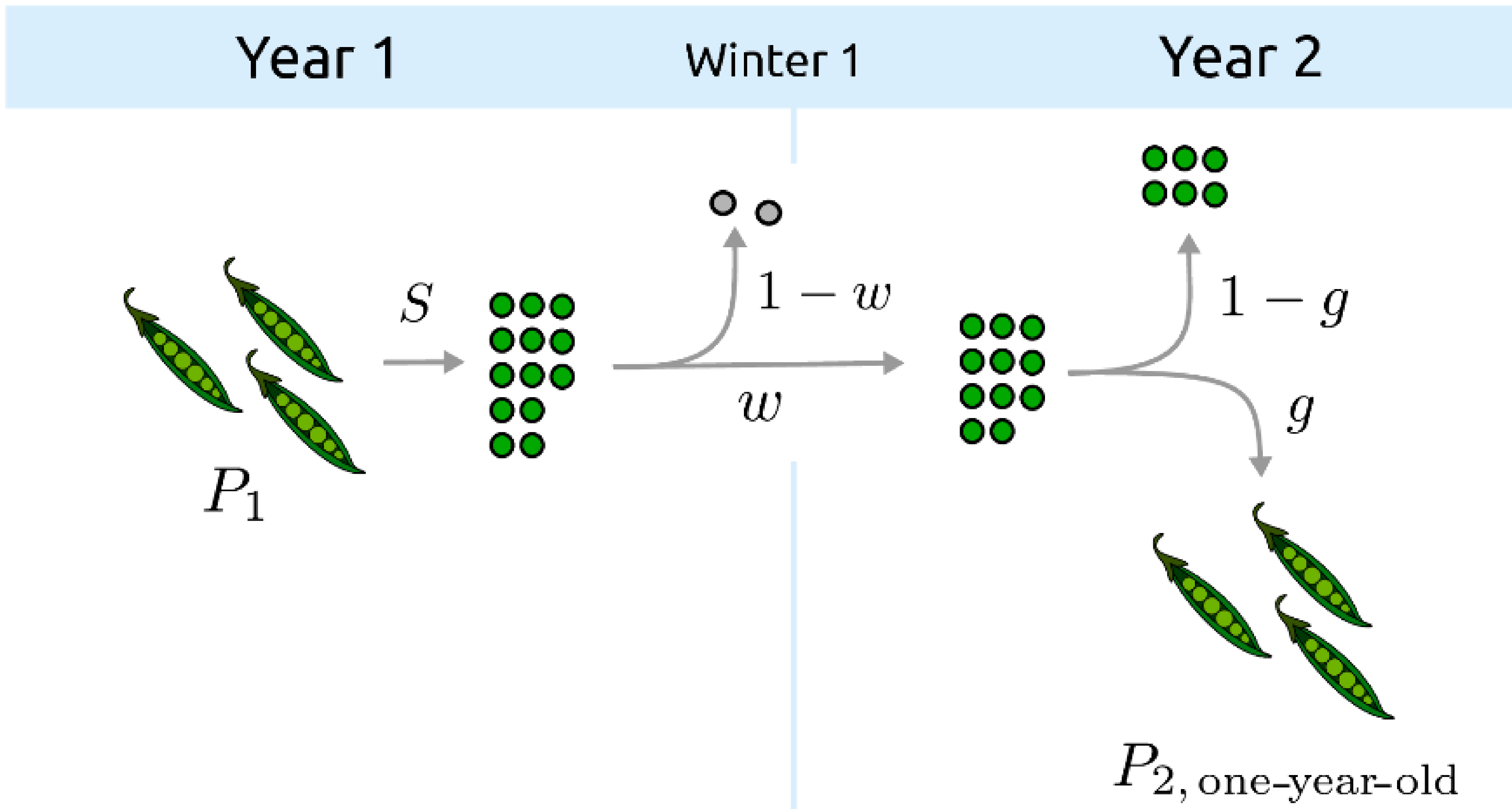
- If $gwS > 1$, the plant population *increases* over successive generations.
- If $gwS = 1$, the plant population *does not change*.
- If $gwS < 1$, the plant population *decreases* over successive generations.

Toårig model

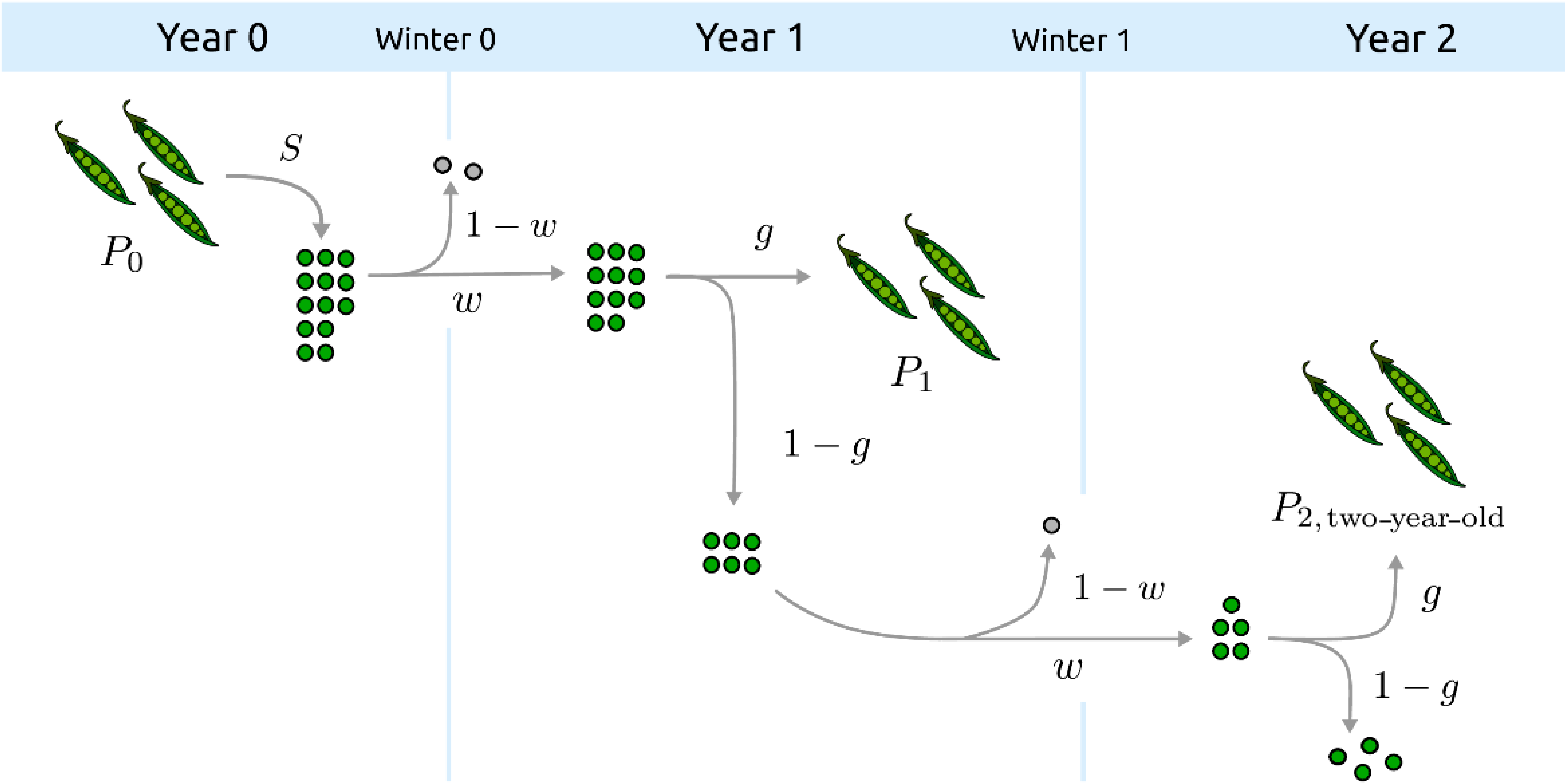
- ny: frø kan overleve to vintre
- S is the number of seeds per plant
- w is the survival rate
- g is the germination rate
- P_n is the number of plants in year n

← →

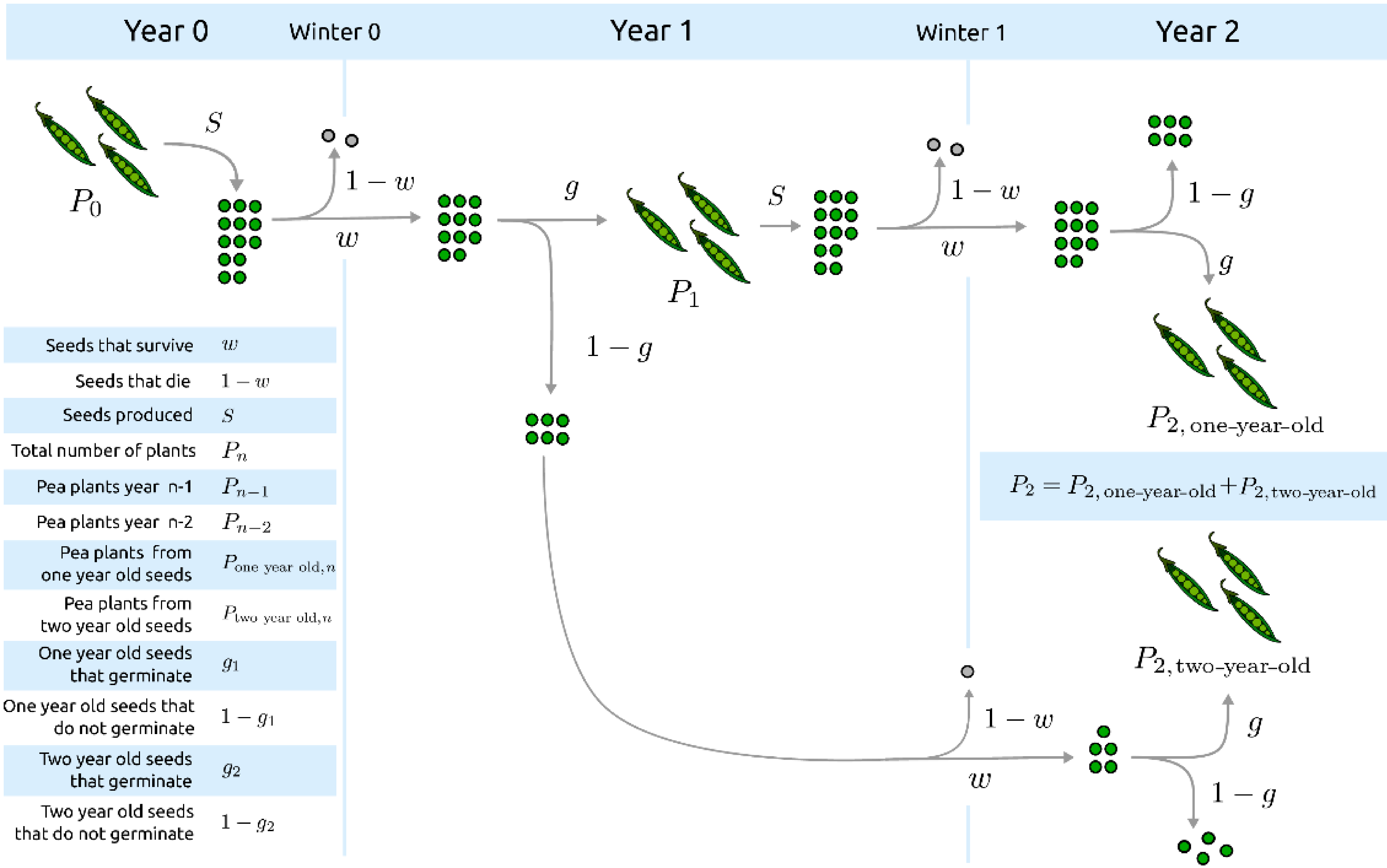
Toårig model



Toårig model



Toårig model



Seeds that survive	w
Seeds that die	$1 - w$
Seeds produced	S
Total number of plants	P_n
Pea plants year n-1	P_{n-1}
Pea plants year n-2	P_{n-2}
Pea plants from one year old seeds	$P_{\text{one year old},n}$
Pea plants from two year old seeds	$P_{\text{two year old},n}$
One year old seeds that germinate	g_1
One year old seeds that do not germinate	$1 - g_1$
Two year old seeds that germinate	g_2
Two year old seeds that do not germinate	$1 - g_2$

Toårig model

For each generation, we calculate the number of pea plants as

$$P_n = P_{n,\text{ one-year-old}} + P_{n,\text{ two-year-old}}$$

Toårig model

One-year-old seeds are found by using the one-year model:

$$P_n = gwSP_{n-1}$$

Toårig model

Remaining seeds:

$$\text{remaining seeds} = (1 - g_n) \times \text{surviving seeds}$$

Toårig model

Multiply the number of remaining seeds by the fraction of seeds that survive the second winter,

$$\text{seeds after the second winter} = w \times \text{remaining seeds}$$

Toårig model

Multiply the number of seeds that survived the second winter with the fraction of seeds that germinate after two years,

$$P_{n,\,\text{two-year-old}} = g_n \times \text{surviving seeds after second winter}$$

← →

Toårig model

```
for n in range(1, N):
    produced_seeds = S * P[n-1]
    surviving_seeds = w * produced_seeds
    plants_1_year = g[n] * surviving_seeds
```

Toårig model

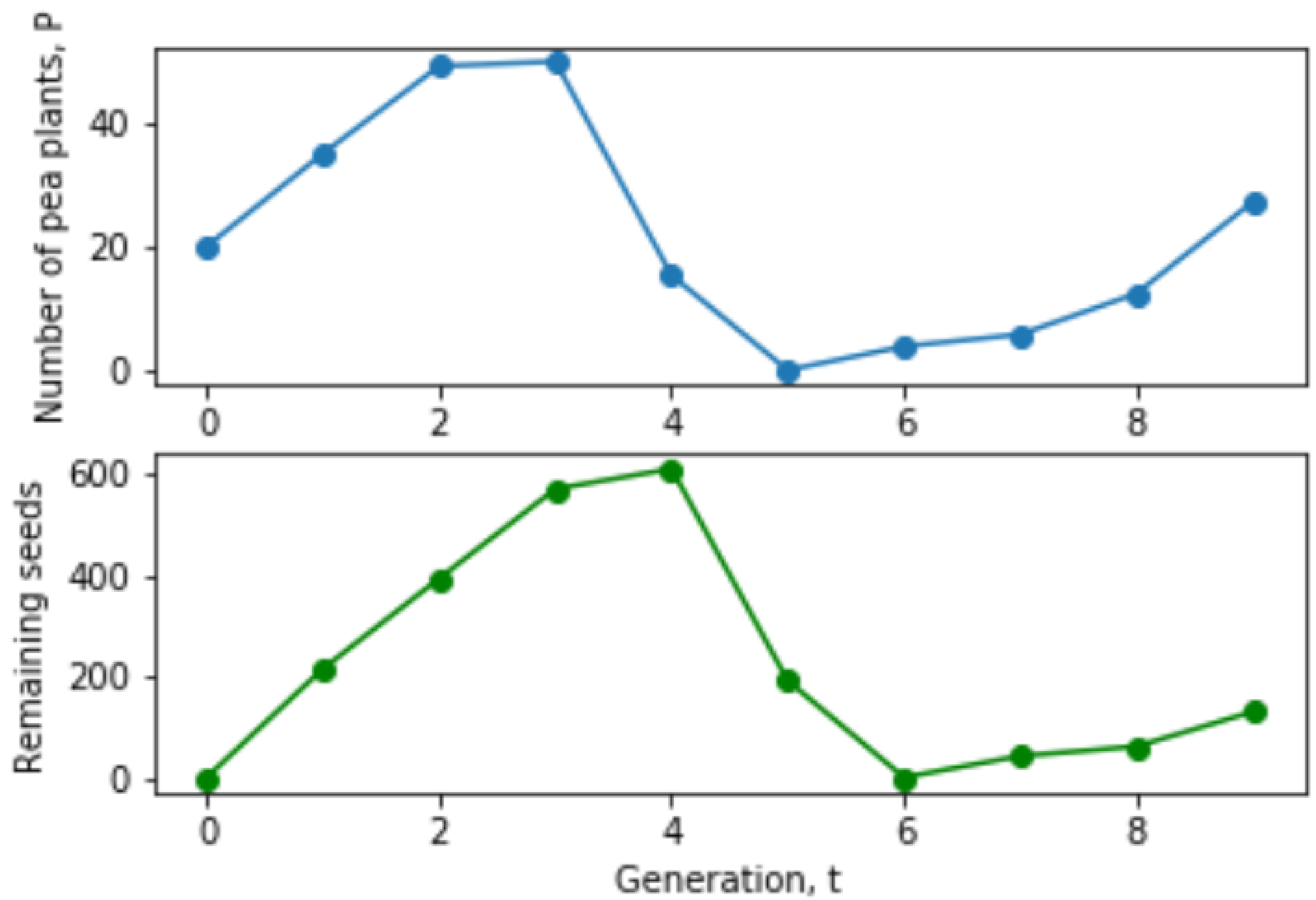
```
for n in range(1, N):
    produced_seeds = S * P[n-1]
    surviving_seeds = w * produced_seeds
    plants_1_year = g[n] * surviving_seeds

    remaining_seeds[n] = (1 - g[n]) * surviving_seeds

    surviving_seeds_2_year = w * remaining_seeds[n-1]
    plants_2_year = g[n] * surviving_seeds_2_year
```

Toårig model

```
for n in range(1, N):
    produced_seeds = S * P[n-1]
    surviving_seeds = w * produced_seeds
    plants_1_year = g[n] * surviving_seeds

    remaining_seeds[n] = (1 - g[n]) * surviving_seeds

    surviving_seeds_2_year = w * remaining_seeds[n-1]
    plants_2_year = g[n] * surviving_seeds_2_year

    P[n] = plants_1_year + plants_2_year
```

Toårig model

Effekt av tørke



Toårig model som differenslikning

If R_n is defined by the number of remaining seeds from the previous generation:

$$R_n = (1 - g_n)wSP_{n-1}$$

Then P_n becomes:

$$\begin{aligned} P_n &= P_{n,\text{ one-year-old}} + P_{n,\text{ two-year-old}} \\ &= g_n wSP_{n-1} + g_n wR_{n-1} \end{aligned}$$

Toårig model som differenslikning

A system of coupled first-order difference equations

$$R_n = (1 - g_n) w S P_{n-1}$$

$$\begin{aligned} P_n &= P_{n,\ \text{one-year-old}} + P_{n,\ \text{two-year-old}} \\ &= g_n w S P_{n-1} + g_n w R_{n-1} \end{aligned}$$

← →

Toårig model som differenslikning

$$R_n = (1 - g_n) w S P_{n-1}$$

Da blir R_{n-1}:

$$R_{n-1} = (1 - g_{n-1}) w S P_{n-2}$$

← →

Toårig model som differenslikning

$$R_{n-1} = (1 - g_{n-1})wSP_{n-2}$$

$$\begin{aligned} P_n &= P_{n,\text{ one-year-old}} + P_{n,\text{ two-year-old}} \\ &= g_n wSP_{n-1} + g_n wR_{n-1} \end{aligned}$$

$$P_n = g_n wSP_{n-1} + g_n w(1 - g_{n-1})wSP_{n-2}$$

Toårig model som differenslikning

Andre ordens differenslikning:

$$P_n = g_n w S P_{n-1} + g_n w (1 - g_{n-1}) w S P_{n-2}$$

Toårig model som differenslikning

and were among the one-year-old seeds that germinated

survived the winter

seeds produced last year

$$P_n = g_n w S P_{n-1} + g_n w (1 - g_{n-1}) w S P_{n-2}$$

seeds produced two years ago

survived first winter

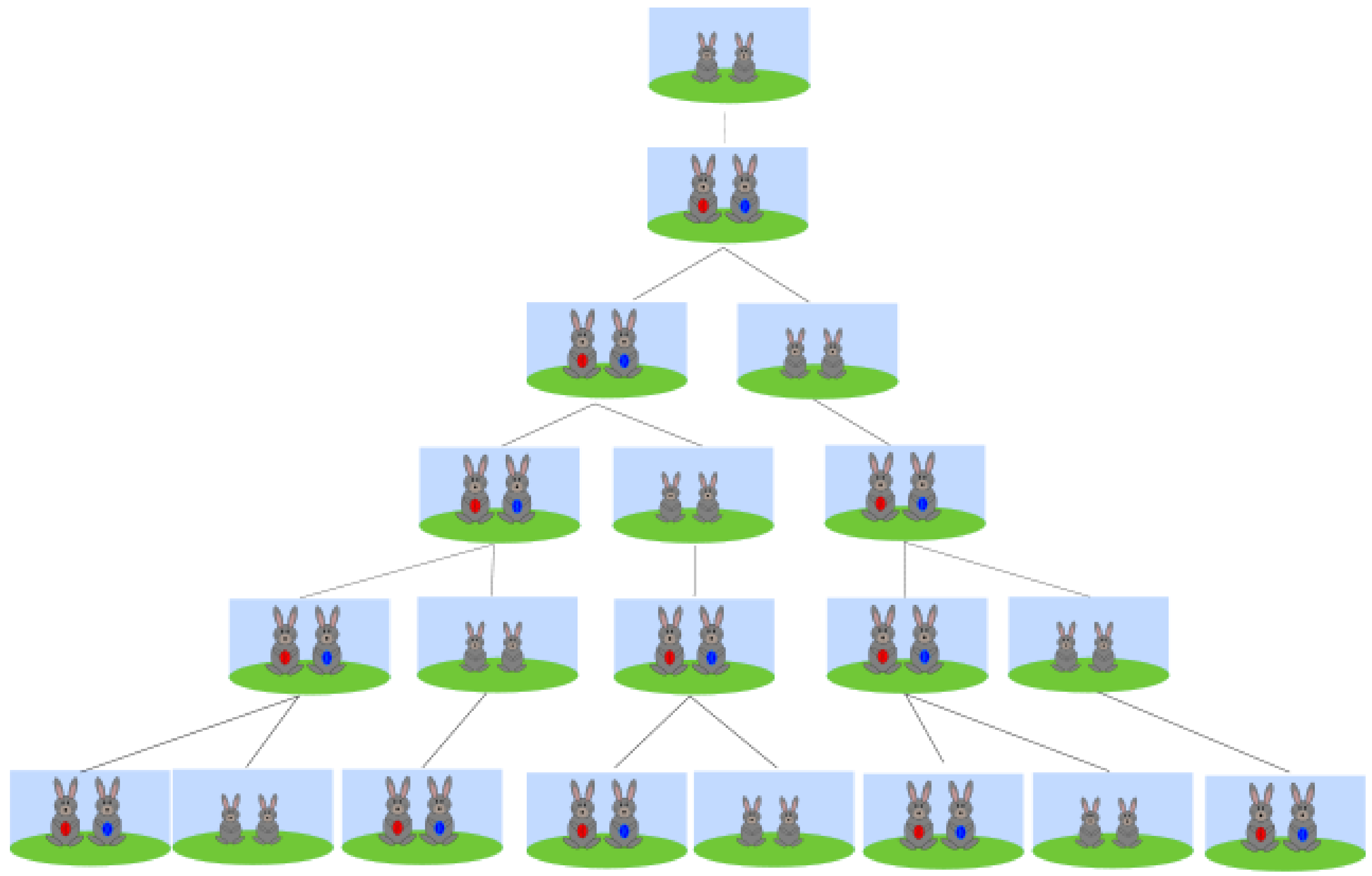
but failed to germinate last year

survived a second winter

and were among the two-year-old seeds that germinated

Harer!

Fibonacci's harer



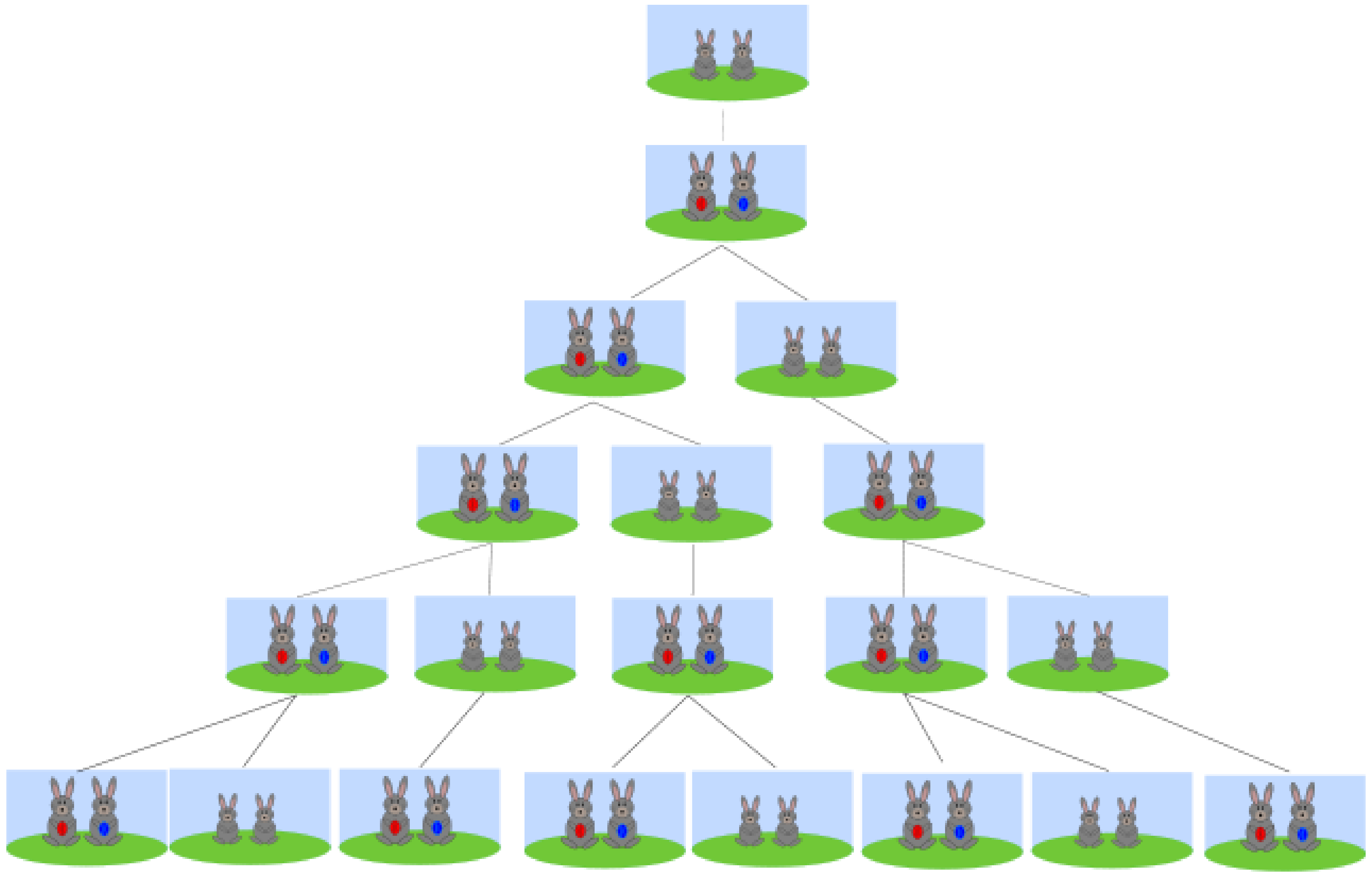
Harer!

Andre-ordrens differenslikninger

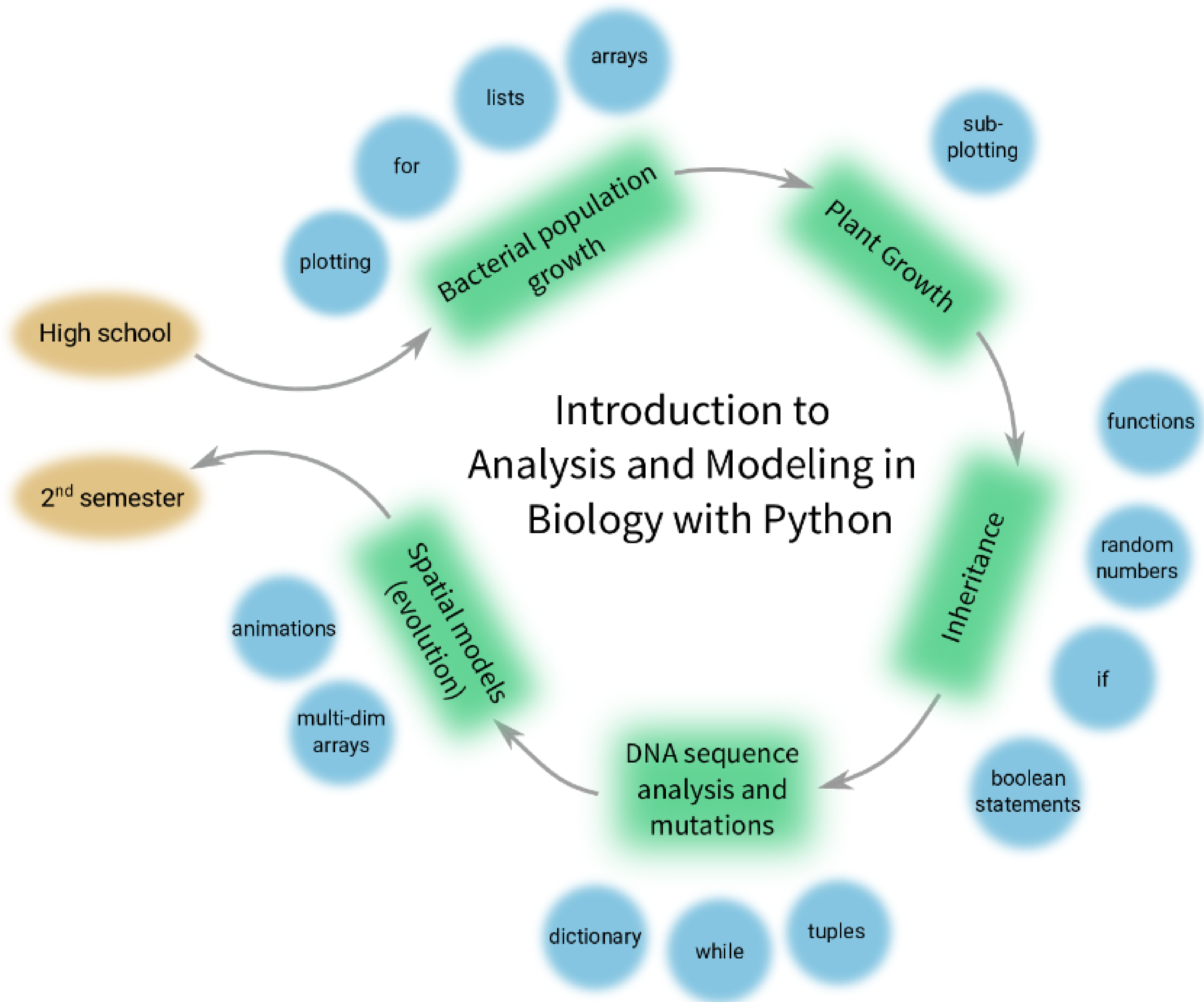
$r_n = r_{n-1} + r_{n-2}$

With initial conditions

$r_0 = 1$ and $r_1 = 1$



Undervisningsplan



Gjøre valg - `if`

```
numbers = [1, 3, 5, 7, 9]

for number in numbers:
  if number > 3:
    print(number)
```

← →

Gjøre valg - `if`

```
numbers = [1, 3, 5, 7, 9]

for number in numbers:
  if number >= 3:
    print(number)
```

← →

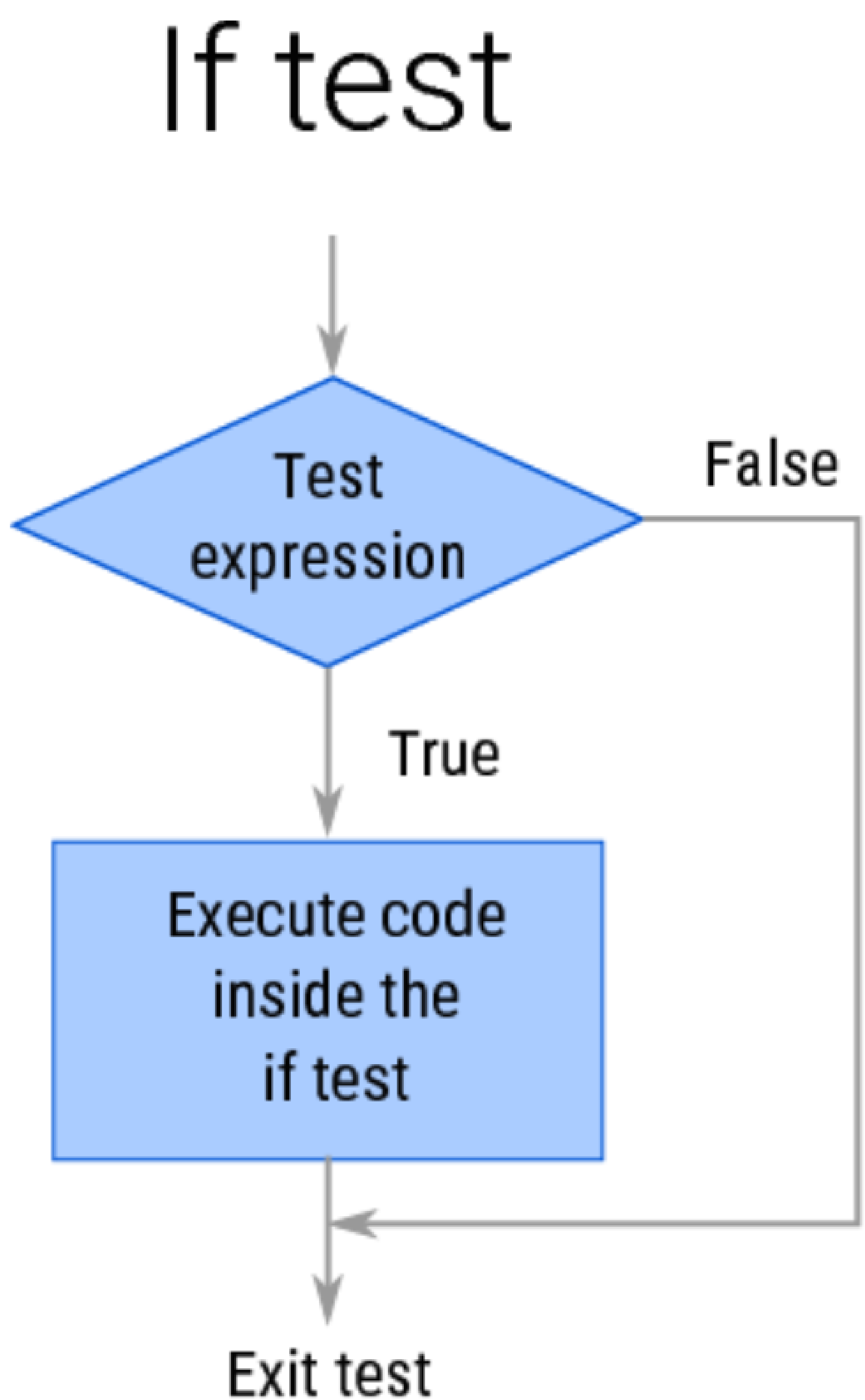
Gjøre valg - `if`

```
numbers = [1, 3, 5, 7, 9]

for number in numbers:
  if number == 3:
    print(number)
```

Gjøre valg - `if`

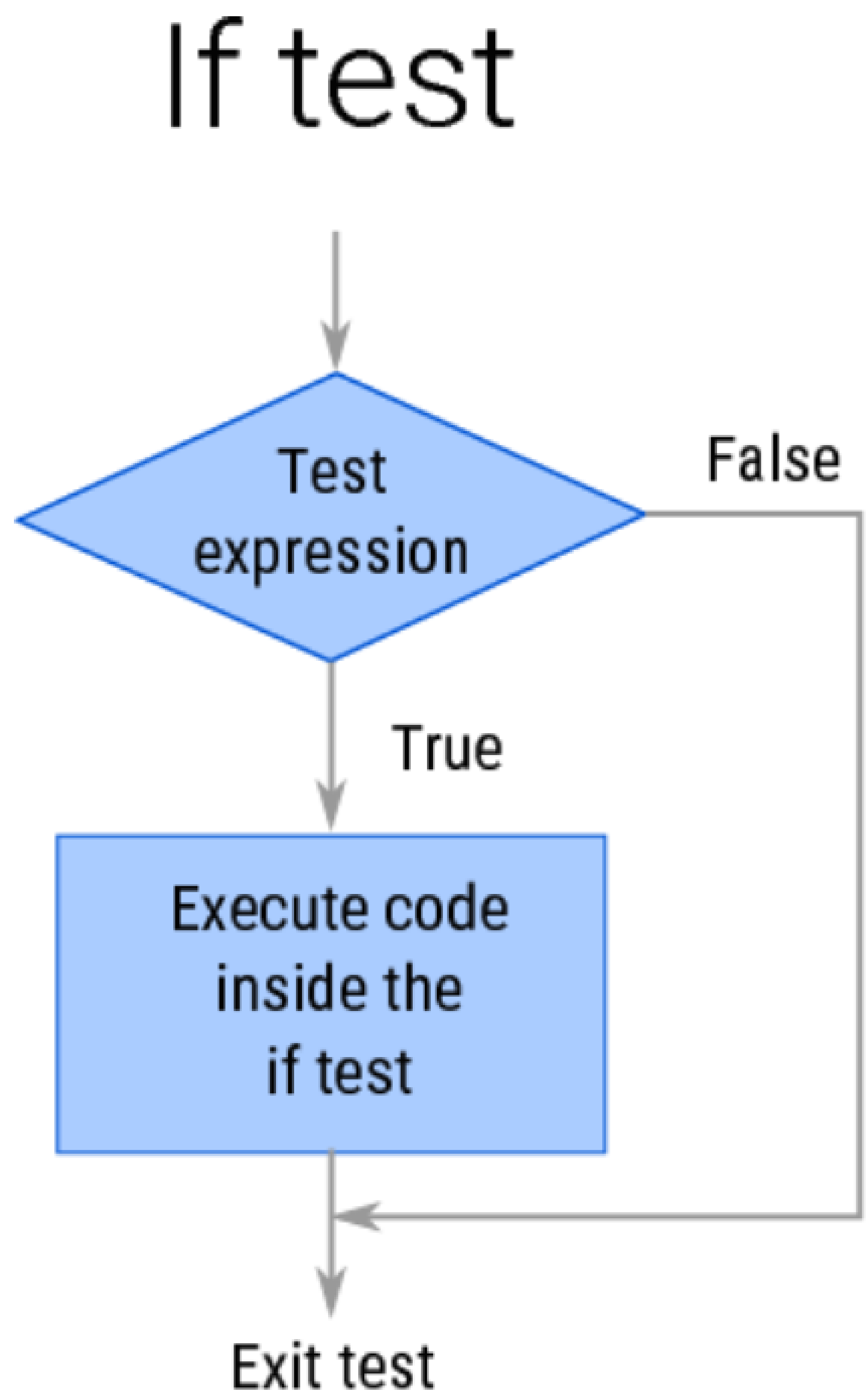
```
names = ['Ola','Kari','Jane','John']

for name in names:
  if name == 'Ola':
    print("Hello, " + name + "!")
```

← →


If test
Test
expression
False
True
Execute code
inside the
if test
Exit test

Gjøre valg - `if`

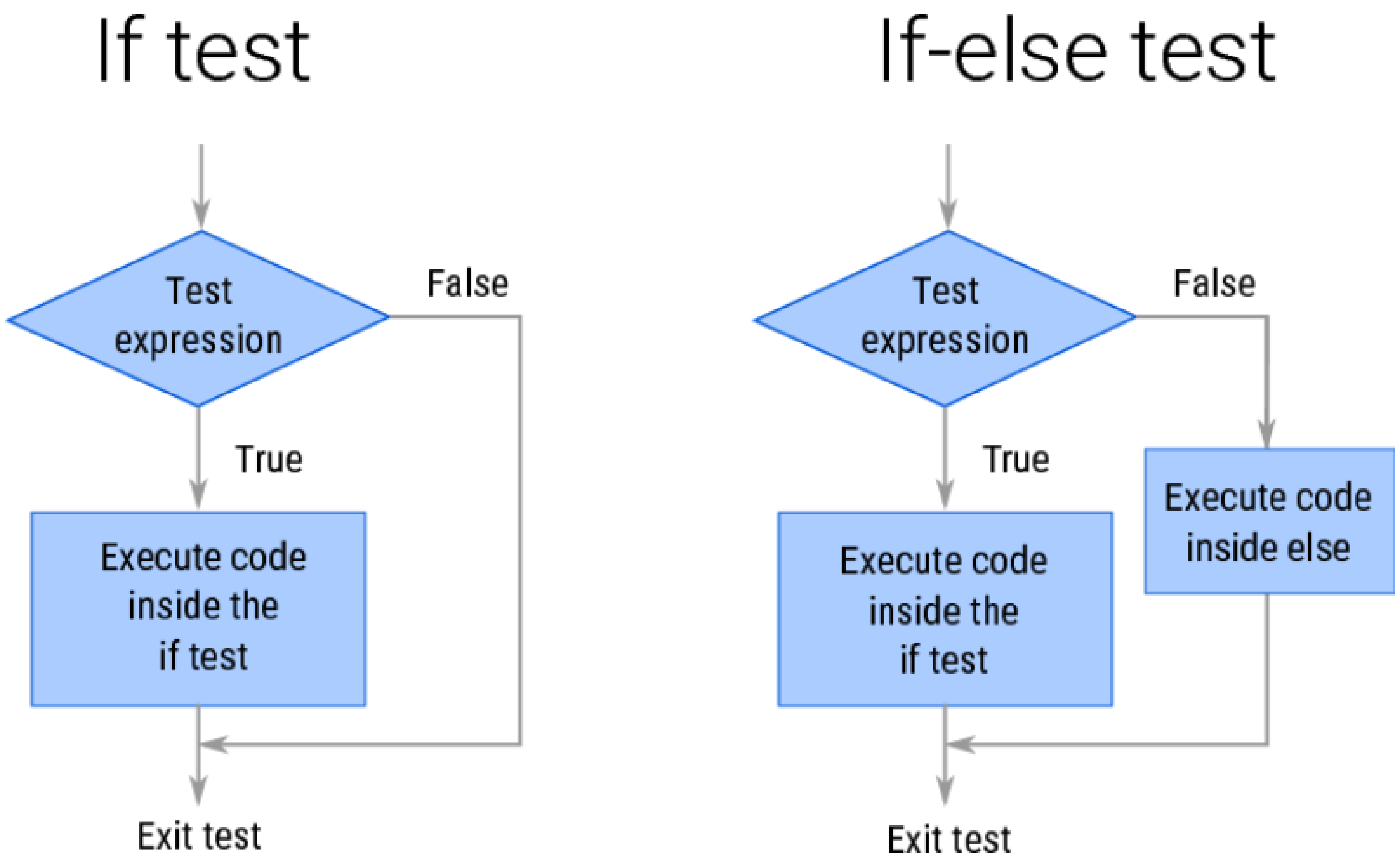
```
if <something is true>:
    <perform an action>
```



Gjøre valg - `if`

```
names = ['Ola','Kari','Jane','John']

for name in names:
  if name == 'Ola':
    print("Hello, " + name + "!")
  else:
    print("Goodbye, " + name + "...")
```

Gjøre valg - `if`



Gjøre valg - `if`

```
names = ['Ola','Kari','Jane','John']

for name in names:
  if name == 'Ola':
    print("Hello, " + name + "!")
```

`==` **is a** *comparison operator*

`=` **is an** *assignment operator*

←

→

Gjøre valg - `if`

```
attendees = ['Ola','Kari','Jane','John']

if 'Kari' in attendees:
    print("Kari is coming!")
```

Gjøre valg - `if`

Expressions that use comparison operators are called *boolean expressions*.

← →

Gjøre valg - `if`

Expressions that use comparison operators are called *boolean expressions*.

`!=` is the comparison operator, while `a != b` is a boolean expression

Gjøre valg - `if`

Syntax	Description
`a == b`	`a` is equal to `b`
`a != b`	`a` is not equal to `b`
`a < b`	`a` is less than `b`
`a > b`	`a` is greater than `b`
`a <= b`	`a` is less than or equal to `b`
`a >= b`	`a` is greater than or equal to `b`
`a in b`	`a` is an element in the list `b`

← →

Utvalgte øvelser

- Exercise 7: Antibiotic levels
- Exercise 6: PCR

Underveisevaluering

Hjelp oss å gjøre kurset bedre!

Gå til menti.com og bruk koden 12 52 65 og svar på spørsmålene der.

Takk på forhånd!