

# Heating

An architect must fit multiple heaters in a large building. In order to make a simulation of the thermoregulation of the building, he must propose a series of heaters. Here, every heater is individually described based on the following information fields: the name of the machine, the current temperature, the minimum and the maximum temperature allowed. In the simulation, the temperature of a certain machine must be increased or decreased, and one must be able at all times to ask the current temperature of every machine.

## Assignment

Make a class `Heating` that supports the following methods:

1. The initializing method `__init__` gets the name of the machine, the current temperature, the minimum and the maximum temperature allowed as parameters. Giving the temperatures is optional. The following standard values are used: 10.0 for the current temperature, 0.0 for the minimum temperature and 100.0 for the maximum temperature.
2. The method `__str__` prints a string representation of the heater. Look at the example below to determine what this string representation should look like. All floating point numbers must be printed with 1 digit after the comma.
3. The method `__repr__` also prints a string representation of the heater. As the method `__str__` is used to obtain the representation of an object that can easily be read for a human user, the `__repr__` prints a representation that can be read by the Python interpreter. The method `__repr__` gives a syntactically correct Python expression, that — when it were to be evaluated — makes an object that is equal to the object that was originally given to `__repr__`.
4. The method `changeTemperature(increment)` that changes the current temperature. As a parameter, the increase in temperature (or decrease in temperature if this is a negative number) is given. This method must however make sure that the temperature is kept within the allowed interval. Should the new temperature be lower than the minimum temperature (or higher than the maximum temperature), this new temperature is equated with the minimum (or maximum) temperature.
5. The method `temperature()` that prints the current temperature as an integer.

## Example

```
>>> machine1 = Heating('radiator kitchen', temperature=20)
>>> machine2 = Heating('radiator living', minimum=15, temperature=18)
>>> machine3 = Heating('radiator bathroom', temperature=22, minimum=18, maximum=28)
>>> print(toestel1)
radiator kitchen: current temperature: 20.0; allowed min: 0.0; allowed max: 100.0
>>> machine2
Heating('radiator living', 18.0, 15.0, 100.0)
>>> machine2.changeTemperature(8)
>>> machine2.temperature()
26.0
>>> machine3.changeTemperature(-5)
>>> machine3
Heating('radiator bathroom', 18.0, 18.0, 28.0)
```

Een architect moet verschillende verwarmingstoestellen plaatsen in het groot gebouw. Om een

simulatie te kunnen maken van de warmteregeling van het gebouw, moet hij een reeks verwarmingstoestellen kunnen voorstellen. Hierbij wordt elk individueel verwarmingstoestel beschreven aan de hand van volgende informatie velden: de naam van het toestel, de huidige temperatuur, de minimum en de maximum toegelaten temperatuur. Binnen de simulatie moet de temperatuur van een bepaald toestel verhoogd of verlaagd kunnen worden, en moet de huidige temperatuur van elke toestel te allen tijden kunnen opgevraagd worden.

## Opgave

Maak een klasse `Verwarming` aan die ondersteuning biedt voor de volgende methoden:

1. De initialisatiemethode `__init__` krijgt als parameters de naam van het toestel, de huidige temperatuur, de minimum en de maximum toegelaten temperatuur. Het meegeven van de temperaturen is optioneel. De volgende standaardwaarden worden gebruikt: 10.0 voor de huidige temperatuur, 0.0 voor de minimumtemperatuur en 100.0 voor de maximumtemperatuur.
2. De methode `__str__` geeft een stringvoorstelling van het verwarmingstoestel terug. Bekijk het voorbeeld hieronder om te bepalen hoe deze stringvoorstelling er moet uitzien. Alle floating point getallen moeten weergegeven worden met 1 cijfer na de komma.
3. De methode `__repr__` geeft eveneens een stringvoorstelling van een verwarmingstoestel terug. Waar de methode `__str__` wordt gebruikt om een voorstelling van een object te bekomen die makkelijk leesbaar is voor een menselijke gebruiker, geeft de `__repr__` een voorstelling terug zoals die kan gelezen worden door de Python interpreter. De methode `__repr__` geeft een syntactisch correcte Python expressie, die — wanneer deze geëvalueerd zou worden — een object aanmaakt dat gelijk is aan het object dat origineel werd doorgegeven aan `__repr__`.
4. De methode `wijzigTemperatuur(increment)` die de huidige temperatuur wijzigt. Als parameter wordt de temperatuursverhoging (of temperatuursverlaging indien dit een negatief getal is) meegegeven. Deze methode moet er wel voor zorgen dat de temperatuur binnen het toegelaten interval blijft. Als de nieuwe temperatuur bijvoorbeeld lager zou zijn dan de minimale temperatuur, dan wordt de nieuwe temperatuur de minimale temperatuur. Analogie voor het overschrijden van de maximale temperatuur.
5. De methode `temperatuur()` die de huidige temperatuur teruggeeft als een reëel getal.

## Voorbeeld

```
>>> toestel1 = Verwarming('radiator keuken', temperatuur=20)
>>> toestel2 = Verwarming('radiator living', minimum=15, temperatuur=18)
>>> toestel3 = Verwarming('radiator badkamer', temperatuur=22, minimum=18, maximum=28)
>>> print(toestel1)
radiator keuken: huidige temperatuur: 20.0; toegelaten min: 0.0; toegelaten max: 100.0
>>> toestel2
Verwarming('radiator living', 18.0, 15.0, 100.0)
>>> toestel2.wijzigTemperatuur(8)
>>> toestel2.temperatuur()
26.0
>>> toestel3.wijzigTemperatuur(-5)
>>> toestel3
Verwarming('radiator badkamer', 18.0, 18.0, 28.0)
```