# Hyperparameter Tuning with mlr3tuning::CHEAT SHEET

## Class Overview

The package provides a set of R6 classes which allow to (a) define general hyperparameter (HP) tuning instances and (b) run algorithms which optimize on these. (a) is called a TuningInstanceSingleCrit or TuningInstanceMultiCrit, which define a blackbox optimization function that maps HP candidate configurations to resampled performance values for arbitrary performance measures.

## Terminators - When to stop

- **evals(n_evals)**: After a given amount of iterations.
- **clock_time(secs, stop_time)**: After a given absolute time.
- **model_time(secs)**: After a given training time.
- **perf_reached(level)**: After a specific performance was reached.
- **stagnation(its, threshold)**: After the performance stagnated for given iterations.

## TuningInstance* - Search Scenario

Evaluator and container for resampled performances of HP configurations during tuning. The main (internal) function eval_batch(xdt) calls benchmark() to evaluate a table of HP configurations. Also stores archive of all evaluated experiments and the final result.

## Executing the Tuning

- `tuner$optimize(instance)`

## Tuner - Search Strategy

Tuning strategy. Generates candidate configurations and passes these to TuningInstance for evaluation until termination. Creation: `trn(.key, ...)`

- `grid_search(resolution, batch_size)` Grid search.
- `random_search(batch_size)` Random search.
- `gensa(smooth, temperature)` Generalized Simulated Annealing.
- `nloptr(algorithm)` Non-linear optimization.
- `design_points(batch_size, design)` User supplied settings.

## AutoTuner - Tune before Train

Wraps learner and performs integrated tuning.

```
at = AutoTuner$new(learner, resampling, measure, terminator, tuner, tune_ps)
```

## Nested Resampling

Resampling the AutoTuner results in nested resampling with an inner and outer loop.

## Example

```
tuner = AutoTuner$new(learner, resampling_inner, measure, evals20, tuner, tune_ps)
```

## AutoTuner

Wraps learner and performs integrated tuning.

```
at = AutoTuner$new(learner, resampling, measure, terminator, tuner, tune_ps)
```

## Terminators

- **$eval_batch(xdt)**

## TuningInstanceMultiCrit

Wraps learner and performs integrated tuning.

```
Optimize hyperpar of RBF SVM on logscale
```

## Example

```
tuner$optimize(instance)
```

## Terminators

- **$eval_batch(xdt)**

## TuningInstanceSingleCrit

Returns all evaluated configurations and their resampling results. Use numres to display HP with (`x_domain`) table applied.

```
result_archive$data()
```

## Terminators

- **$eval_batch(xdt)**

## TuningInstanceMultiCrit

Returns all evaluated configurations and their resampling results. Use numres to display HP with (`x_domain`) table applied.

```
result_archive$data()
```

## Terminators

- **$eval_batch(xdt)**

## TuningInstanceSingleCrit

Returns all evaluated configurations and their resampling results. Use numres to display HP with (`x_domain`) table applied.

```
result_archive$data()
```