

# SLiM Reference Sheet

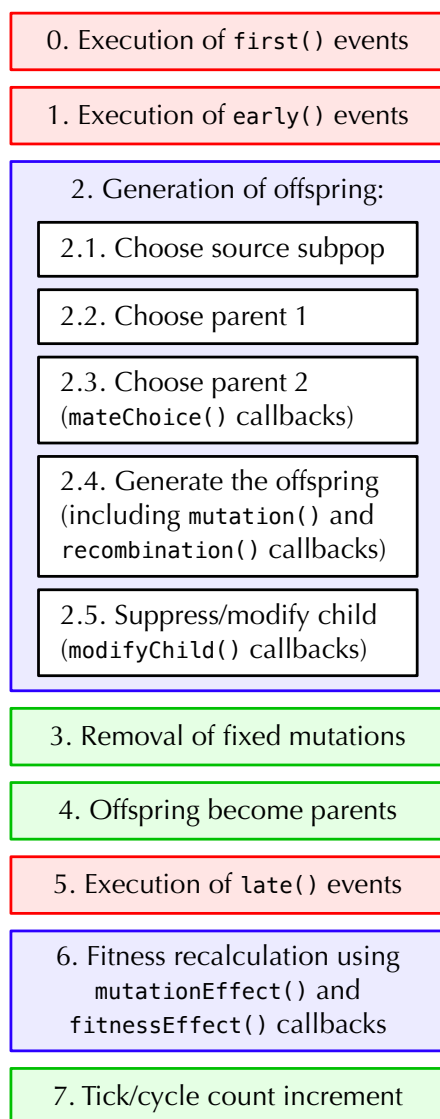
14 September 2022

```
slim -version | -usage | -help | -testEidos | -testSLiM |
[-seed <seed>] [-time] [-mem] [-Memhist] [-long [<l>]] [-x] [-define <def>] <script file>
```

**Types:** N:NULL, l:logical, i:integer, f:float, n:numeric, s:string, o<X>:object of class X

(ret-type)functionName(params) { ... }	user-defined function
[id] initialize() { ... }	initialize() callback
[id] [t1 [: t2]] first()   early()   late() { ... }	Eidos events
[id] [t1 [: t2]] fitnessEffect([<subpopId>]) { ... }	fitnessEffect() callback
[id] [t1 [: t2]] interaction(<intTypeId> [, <subpopId>]) { ... }	interaction() callback
[id] [t1 [: t2]] mateChoice([<subpopId>]) { ... } (WF)	mateChoice() callback
[id] [t1 [: t2]] modifyChild([<subpopId>]) { ... }	modifyChild() callback
[id] [t1 [: t2]] mutationEffect(<mutTypeId> [, <subpopId>]) { ... }	mutationEffect() callback
[id] [t1 [: t2]] recombination([<subpopId>]) { ... }	recombination() callback
[id] [t1 [: t2]] mutation([<mutTypeId> [, <subpopId>]]) { ... }	mutation() callback
[id] [t1 [: t2]] reproduction([<subpopId> [, <sex>]]) { ... } (nonWF)	reproduction() callback
[id] [t1 [: t2]] survival([<subpopId>]) { ... } (nonWF)	survival() callback

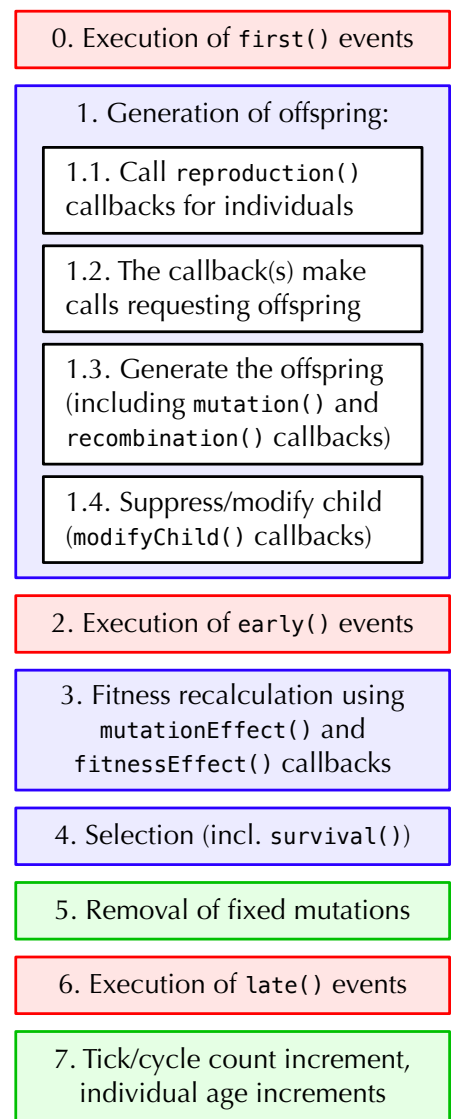
## The WF model's tick cycle



## Multispecies execution:

- definition order
- species declaration order
- undefined order

## The nonWF model's tick cycle



\* : cycle stage present in only one of the model types

## Initialization functions (callable only from initialize() callbacks):

```
(integer$)initializeAncestralNucleotides(is sequence) (nuc)
(void)initializeGeneConversion(n$ nonCrossoverFraction, n$ meanLength,
    n$ simpleConversionFraction, [n$ bias])
(o<GElement>)initializeGenomicElement(io<GType> genomicElementType, i start, i end)
(o<GType>$)initializeGenomicElementType(is$ id, io<MutType> mutationTypes, n proportions,
    [Nf mutationMatrix])
(void)initializeHotspotMap(numeric multipliers, [Ni ends], [s$ sex]) (nuc)
(o<IntType>$)initializeInteractionType(is$ id, s$ spatiality, [l$ reciprocal],
    [n$ maxDistance], [s$ sexSegregation])
(void)initializeMutationRate(n rates, [Ni ends], [s$ sex])
(o<MutType>$)initializeMutationType(is$ id, n$ dominanceCoeff, s$ distributionType, ...)
(o<MutType>$)initializeMutationTypeNuc(is$ id, n$ dominanceCoeff, s$ distributType, ...) (nuc)
(void)initializeRecombinationRate(n rates, [Ni ends], [s$ sex])
(void)initializeSex(s$ chromosomeType)
(void)initializeSLiMModelType(s$ modelType)
(void)initializeSLiMOptions([l$ keepPedigrees], [s$ dimensionality], [s$ periodicity],
    [i$ mutationRuns], [l$ preventIncidentalSelfing], [l$ nucleotideBased],
    [l$ randomizeCallbacks])
(void)initializeSpecies([i$ tickModulo], [i$ tickPhase], [s$ avatar], [s$ color])
(void)initializeTreeSeq([l$ recordMutations], [Nif$ simplificationRatio],
    [Ni$ simplificationInterval], [l$ checkCoalescence], [l$ runCrosschecks],
    [l$ retainCoalescentOnly], [Ns$ timeUnit])
```

## SLiM callbacks:

### mutationEffect():

```
mut (o<Mutation>$)
homozygous (l$)
effect (f$)
individual (o<Ind>$)
subpop (o<Subpop>$)
float$: mutation effect
```

### reproduction(): (nonWF)

```
individual (o<Ind>$)
subpop (o<Subpop>$)
void: (no return)
```

### mutation():

```
mut (o<Mutation>$)
genome (o<Genome>$)
element (o<GElement>$)
originalNuc (i$)
parent (o<Ind>$)
subpop (o<Subpop>$)
T: accept the mutation
F: reject the mutation
o<Mut>$: use a substitute
```

### mateChoice(): (WF)

```
individual (o<Ind>$)
subpop (o<Subpop>$)
sourceSubpop (o<Subpop>$)
weights (f)
NULL: use default mate choice
float: new mating weights
float(0): no suitable mate
o<Ind>$: the chosen mate
```

### modifyChild():

```
child (o<Ind>$)
parent1 (o<Ind>$)
isCloning (l$)
isSelfing (l$)
parent2 (o<Ind>$)
subpop (o<Subpop>$)
sourceSubpop (o<Subpop>$)
T: accept the proposed child
F: reject the proposed child
```

### SLiMEidosBlock (SEBlock):

Superclass: Dictionary

```
active <=> (i$)
end => (i$)
id => (i$)
source => (s$)
speciesSpec => (o<Species>$)
start => (i$)
tag <=> (i$)
ticksSpec => (o<Species>$)
type => (s$)
```

## SLiM globals:

```
community (o<Community>$)
sim, ... (o<Species>$)
g1, ... (o<GType>$)
i1, ... (o<IntType>$)
m1, ... (o<MutType>$)
p1, ... (o<Subpop>$)
s1, ... (o<SEBlock>$)
self (o<SEBlock>$)
```

### recombination():

```
individual (o<Ind>$)
genome1 (o<Genome>$)
genome2 (o<Genome>$)
subpop (o<Subpop>$)
breakpoints (i)
T: breakpoints changed
F: breakpoints unchanged
```

### interaction():

```
distance (f$)
strength (f$)
exerter (o<Ind>$)
receiver (o<Ind>$)
float$: interaction strength
```

### survival(): (nonWF)

```
individual (o<Ind>$)
subpop (o<Subpop>$)
surviving (l$)
fitness (f$)
draw (f$)
T: individual survives
F: individual dies
NULL: use SLiM's decision
o<Subpop>$: move to subpop
```

### fitnessEffect():

```
individual (o<Ind>$)
subpop (o<Subpop>$)
float$: fitness effect
```

## Species:

Superclass: *Dictionary*

```
avatar => (s$)
chromosome => (o<Chromosome>$)
chromosomeType => (s$)
color => (s$)
cycle <=> (i$)
description <=> (s$)
dimensionality => (s$)
genomicElementTypes => (o<GEType>)
id => (i$)
mutationTypes => (o<MutType>)
mutations => (o<Mut>)
name => (s$)
nucleotideBased => (l$) (nuc)
periodicity => (s$)
scriptBlocks => (o<SEBlock>)
sexEnabled => (l$)
subpopulations => (o<Subpop>)
substitutions => (o<Substitution>)
tag <=> (i$)

- (o<Subpop>$)addSubpop(is$ subpopID, i$ size, [f$ sexRatio], [l$ haploid])
- (o<Subpop>$)addSubpopSplit(is$ subpopID, i$ size, io<Subpop>$ sourceSubpop,
  [f$ sexRatio]) (WF)
- (i$)countOfMutationsOfType(io<MutType>$ mutType)
- (o<Ind>)individualsWithPedigreeIDs(i pedigreeIDs, [Nio<Subpop> subpops])
- (void)killIndividuals(o<Individual> individuals) (nonWF)
- (i)mutationCounts(Nio<Subpop> subpops, [No<Mut> mutations])
- (f)mutationFrequencies(Nio<Subpop> subpops, [No<Mut> mutations])
- (o<Mut>)mutationsOfType(io<MutType>$ mutType)
- (void)outputFixedMutations([Ns$ filePath], [l$ append])
- (void)outputFull([Ns$ filePath], [l$ binary], [l$ append], [l$ spatialPositions], [l$ ages],
  [l$ ancestralNucleotides], [l$ pedigreeIDs])
- (void)outputMutations(o<Mut> mutations, [Ns$ filePath], [l$ append])
- (i$)readFromPopulationFile(s$ filePath, [No$ subpopMap])
- (void)recalculateFitness([Ni$ tick])
- (o<SEBlock>$)register[FitnessEffect/MateChoice (WF)/ModifyChild/Recombination/
  Survival]Callback(Nis$ id, s$ source, [Nio<Subpop>$ subpop], [Ni$ start], [Ni$ end])
- (o<SEBlock>$)registerMutationCallback(Nis$ id, s$ source, [Nio<MutType>$ mutType],
  [Nio<Subpop>$ subpop], [Ni$ start], [Ni$ end])
- (o<SEBlock>$)registerMutationEffectCallback(Nis$ id, s$ source, io<MutType>$ mutType,
  [Nio<Subpop>$ subpop], [Ni$ start], [Ni$ end])
- (o<SEBlock>$)registerReproductionCallback(Nis$ id, s$ source, [Nio<Subpop>$ subpop],
  [Ns$ sex], [Ni$ start], [Ni$ end]) (nonWF)
- (void)simulationFinished(void)
- (void)skipTick(void)
- (o<Mut>)subsetMutations([No<Mut>$ exclude], [Nio<MutType>$ mutType], [Ni$ position],
  [Nis$ nucleotide], [Nl$ tag], [Nl$ id])
- (l$)treeSeqCoalesced(void)
- (void)treeSeqOutput(s$ path, [l$ simplify], [l$ includeModel], [No$ metadata])
- (void)treeSeqRememberIndividuals(o<Ind> individuals, [l$ permanent])
- (void)treeSeqSimplify(void)
```

Effects of mutations:

no mutation present	1
heterozygote	$1 + h*s$
homozygote	$1 + s$

$s = \text{mut.selectionCoeff}$   
 $h = \text{mutType.dominanceCoeff}$

## GenomicElement (GElement):

Superclass: *Object*

```
endPosition => (i$)
genomicElementType => (o<GEType>$)
startPosition => (i$)
tag <=> (i$)

- (void)setGenomicElementType(io<GEType>$ genomicElementType)
```

## Subpopulation (Subpop):

Superclass: *Dictionary*

```
cloningRate => (f) (WF)
description <=> (s$)
firstMaleIndex => (i$)
fitnessScaling <=> (f$)
genomes => (o<Genome>)
genomesNonNull => (o<Genome>)
id => (i$)
immigrantSubpopFractions => (f) (WF)
immigrantSubpopIDs => (i) (WF)
individualCount => (i$)
individuals => (o<Ind>)
lifetimeReproductiveOutput => (i$)
lifetimeReproductiveOutputF => (i$)
lifetimeReproductiveOutputM => (i$)
name <=> (s$)
selfingRate => (f$) (WF)
sexRatio => (f$) (WF)
spatialBounds => (f)
species => (o<Species>$)
tag <=> (i$)

- (No<Ind>$)addCloned(o<Ind>$ parent) (nonWF)
- (No<Ind>$)addCrossed(o<Ind>$ parent1, o<Ind>$ parent2, [Nfs$ sex]) (nonWF)
- (No<Ind>$)addEmpty([Nfs$ sex], [Nl$ genome1Null], [Nl$ genome2Null]) (nonWF)
- (No<Ind>$)addRecombinant(No<Genome>$ strand1, No<Genome>$ strand2, Ni breaks1,
  No<Genome>$ strand3, No<Genome>$ strand4, Ni breaks2, [Nfs$ sex]) (nonWF)
- (No<Ind>$)addSelfed(o<Ind>$ parent) (nonWF)
- (f)cachedFitness(Ni indices)
- (void)configureDisplay([Nf center], [Nf$ scale], [Ns$ color])
- (void)defineSpatialMap(s$ name, s$ spatiality, n values, [l$ interpolate],
  [Nif valueRange], [Ns colors])
- (void)outputMSSample(i$ sampleSize, [l$ replace], [s$ requestedSex],
  [Ns$ filePath], [l$ append], [l$ filterMonomorphic])
- (void)outputSample(i$ sampleSize, [l$ replace], [s$ requestedSex],
  [Ns$ filePath], [l$ append])
- (void)outputVCFsample(i$ sampleSize, [l$ replace], [s$ requestedSex],
  [l$ outputMultiallelics], [Ns$ filePath], [l$ append])
- (l)pointInBounds(f point)
- (f)point[Periodic|Reflected|Stopped](f point)
- (f)pointUniform([i$ n])
- (void)removeSubpopulation(void) (nonWF)
- (void)sampleIndividuals(i$ size, [l$ replace], [No<Ind>$ exclude], [Ns$ sex], [Ni$ tag],
  [Ni$ minAge], [Ni$ maxAge], [Nl$ migrant])
- (void)setCloningRate(n rate) (WF)
- (void)setMigrationRates(io<Subpop> sourceSubpops, n rates) (WF)
- (void)setSelfingRate(n$ rate) (WF)
- (void)setSexRatio(f$ sexRatio) (WF)
- (void)setSpatialBounds(n bounds)
- (void)setSubpopulationSize(i$ size) (WF)
- (s)spatialMapColor(s$ name, n value)
- (o<Image>$) spatialMapImage(s$ name, [Ni$ width], [Ni$ height], [l$ centers], [l$ color])
- (f)spatialMapValue(s$ name, f point)
- (o<Ind>)subsetIndividuals([No<Ind>$ exclude], [Ns$ sex], [Ni$ tag], [Ni$ minAge],
  [Ni$ maxAge], [Nl$ migrant])
- (void)takeMigrants(o<Ind> migrants) (nonWF)
```

## SLiMgui: (in SLiMgui only)

Superclass: *Dictionary*

```
pid => (i$)
- (void)openDocument(s$ filePath)
- (void)pauseExecution(void)
```

SLiMgui quick help:  
opt-click on keyword

Code completion:  
escape (ESC)

## Individual (Ind):

*Superclass: Dictionary*

```
age <=> (i$) (nonWF)
color <=> (s$)
fitnessScaling <=> (f$)
genomes => (o<Genome>)
genomesNonNull => (o<Genome>)
genome1 => (o<Genome>$)
genome2 => (o<Genome>$)
index => (i$)
migrant => (l$)
pedigreeID => (i$)
pedigreeParentIDs => (i)
pedigreeGrandparentIDs => (i)
reproductiveOutput => (i$)
sex => (s$)
spatialPosition => (f)
subpopulation => (o<Subpop>$)
tag <=> (i$)
tagF <=> (f$)
uniqueMutations => (o<Mut>)
x <=> (f$)
y <=> (f$)
z <=> (f$)
- (l)containsMutations(o<Mut> mutations)
- (i$)countOfMutationsOfType(io<MutType>$ mutType)
- (f)relatedness(o<Ind> individuals)
+ (void)setSpatialPosition(f position)
- (f$)sumOfMutationsOfType(io<MutType>$ mutType)
- (o<Mut>)uniqueMutationsOfType(io<MutType>$ mutType)
```

## Chromosome:

*Superclass: Dictionary*

```
colorSubstitution <=> (s$)
geneConversionEnabled => (l$)
geneConversionGCbias => (f$)
geneConversionNonCrossoverFraction => (f$)
geneConversionMeanLength => (f$)
geneConversionSimpleConversionFraction => (f$)
genomicElements => (o<GElement>)
hotspotEndPositions[F|M] => (i) (nuc)
hotspotMultipliers[F|M] => (f) (nuc)
lastPosition => (i$)
mutationEndPositions[F|M] => (i)
mutationRates[F|M] => (f)
overallMutationRate[F|M] => (f$)
overallRecombinationRate[F|M] => (f$)
recombinationEndPositions[F|M] => (i)
recombinationRates[F|M] => (f)
species => (o<Species>$)
tag <=> (i$)
- (is)ancestralNucleotides([Ni$ start], [Ni$ end], [s$ format]) (nuc)
- (integer)drawBreakpoints([No<Ind>$ parent], [Ni$ n])
- (integer$)setAncestralNucleotides(is sequence) (nuc)
- (void)setGeneConversion(n$ nonCrossoverFraction, n$ meanLength,
  n$ simpleConversionFraction, [n$ bias])
- (void)setHotspotMap(n multipliers, [Ni ends], [s$ sex]) (nuc)
- (void)setMutationRate(n rates, [Ni ends], [s$ sex])
- (void)setRecombinationRate(n rates, [Ni ends], [s$ sex])
```

## Mutation (Mut):

*Superclass: Dictionary*

```
id => (i$)
isFixed => (l$)
isSegregating => (l$)
mutationType => (o<MutType>$)
nucleotide <=> (s$) (nuc)
nucleotideValue <=> (i$) (nuc)
originTick => (i$)
position => (i$)
selectionCoeff => (f$)
subpopID <=> (i$)
tag <=> (i$)
- (void)setMutationType(io<MutType>$ mutType)
- (void)setSelectionCoeff(f$ selectionCoeff)
```

## Substitution:

*Superclass: Dictionary*

```
id => (i$)
fixationTick => (i$)
mutationType => (o<MutType>$)
nucleotide <=> (s$) (nuc)
nucleotideValue <=> (i$) (nuc)
originTick => (i$)
position => (i$)
selectionCoeff => (f$)
subpopID <=> (i$)
tag <=> (i$)
```

## MutationType (MutType):

*Superclass: Dictionary*

```
color <=> (s$)
colorSubstitution <=> (s$)
convertToSubstitution <=> (l$)
distributionParams => (f)
distributionType => (s$)
dominanceCoeff <=> (f$)
id => (i$)
mutationStackGroup <=> (i$)
mutationStackPolicy <=> (s$)
nucleotideBased => (l$) (nuc)
species => (o<Species>$)
tag <=> (i$)
- (float)drawSelectionCoefficient(i$ n)
- (void)setDistribution(s$ distType, ...)
```

## Genome:

Superclass: *Object*

```
genomePedigreeID => (i$)
genomeType => (s$)
individual => (o<Ind>$)
isNullGenome => (l$)
mutations => (o<Mut>)
tag <=> (i$)

+ (void)addMutations(o<Mut> mutations)
+ (o<Mut>)addNewDrawnMutation(io<MutType> mutationType, i position,
    [Nio<Subpop> originSubpop], [Nis nucleotide])
+ (o<Mut>)addNewMutation(io<MutType> mutationType, n selectionCoeff, i position,
    [Nio<Subpop> originSubpop], [Nis nucleotide])
- (l$)containsMarkerMutation(io<MutType>$ mutType, i$ position, [l$ returnMutation])
- (l)containsMutations(o<Mut> mutations)
- (i$)countOfMutationsOfType(io<MutType>$ mutType)
+ (i)mutationCountsInGenomes(No<Mut> mutations)
+ (f)mutationFrequenciesInGenomes(No<Mut> mutations)
- (o<Mut>)mutationsOfType(io<MutType>$ mutType)
- (is)nucleotides([Ni$ start], [Ni$ end], [s$ format])
+ (void)output([Ns$ filePath], [l$ append])
+ (void)outputMS([Ns$ filePath], [l$ append], [l$ filterMonomorphic])
+ (void)outputVCF([Ns$ filePath], [l$ outputMultiallelics], [l$ append])
- (i)positionsOfMutationsOfType(io<MutType>$ mutType)
+ (o<Mut>)readFromMS(s$ filePath, io<MutType>$ mutationType)
+ (o<Mut>)readFromVCF(s$ filePath, [Nio<MutType>$ mutationType])
+ (void)removeMutations([No<Mut> mutations], [l$ substitute])
- (f$)sumOfMutationsOfType(io<MutType>$ mutType)
```

## InteractionType (IntType):

Superclass: *Dictionary*

```
id => (i$)
maxDistance <=> (f$)
reciprocal => (l$)
sexSegregation => (s$)
spatiality => (s$)
tag <=> (i$)

- (f)clippedIntegral(No<Ind> individuals)
- (f)distance(o<Ind>$ receiver, [No<Ind> exerters])
- (f)distanceFromPoint(f point, o<Ind> exerters)
- (o<Ind>)drawByStrength(o<Ind>$ receiver, [i$ count], [No<Subpop>$ exorterSubpop])
- (void)evaluate(io<Subpop> subpops)
- (i)interactingNeighborCount(o<Ind> receivers, [No<Subpop>$ exorterSubpop])
- (f)interactionDistance(o<Ind> receiver, [No<Ind> exerters])
- (f)localPopulationDensity(o<Ind> receivers, [No<Subpop>$ exorterSubpop])
- (o<Ind>)nearestInteractingNeighbors(o<Ind>$ receiver, [i$ count],
    [No<Subpop>$ exorterSubpop])
- (o<Ind>)nearestNeighbors(o<Ind>$ receiver, [i$ count], [No<Subpop>$ exorterSubpop])
- (o<Ind>)nearestNeighborsOfPoint(f point, io<Subpop>$ exorterSubpop, [i$ count])
- (i)neighborCount(o<Ind> receivers, [No<Subpop>$ exorterSubpop])
- (i)neighborCountOfPoint(o<Ind> receivers, [No<Subpop>$ exorterSubpop])
- (void)setInteractionFunction(s$ functionType, ...)
- (f)strength(o<Ind>$ receiver, [No<Ind> exerters])
- (f)totalOfNeighborStrengths(o<Ind> receivers, [No<Subpop>$ exorterSubpop])
- (void)unevaluate(void)
```

## GenomicElementType (GEType):

Superclass: Dictionary

```
color <=> (s$)
id => (i$)
mutationFractions => (f)
mutationMatrix => (f) (nuc)
mutationTypes => (o<MutType>)
species => (o<Species> $)
tag <=> (i$)

- (void)setMutationFractions(io<MutType> mutationTypes, n proportions)
- (void)setMutationMatrix(f mutationMatrix) (nuc)
```

## LogFile:

Superclass: Dictionary

```
filePath => (s$)
logInterval => (i$)
precision <=> (i$)
tag <=> (i$)

- (void)addCustomColumn(s$ columnName, s$ source, [* context])
- (void)addCycle(void)
- (void)addCycleStage(void)
- (void)addMeanSDColumns(s$ columnName, s$ source, [* context])
- (void)addPopulationSexRatio(void)
- (void)addPopulationSize(void)
- (void)addSubpopulationSexRatio(io<Subpop>$ subpop)
- (void)addSubpopulationSize(io<Subpop>$ subpop)
- (void)addSuppliedColumn(s$ columnName)
- (void)flush(void)
- (void)logRow(void)
- (void)setFilePath(s$ filePath, [Ns initialContents], [l$ append], [Nl$ compress], [Ns$ sep])
- (void)setLogInterval([Ni$ logInterval])
- (void)setSuppliedValue(s$ columnName, +$ value)
- (void)willAutolog(void)
```

## SLiM population-genetics utilities:

```
- (f$)calcFST(o<Genome> genomes1, o<Genome> genomes2, [No<Mut> muts], [Ni$ start], [Ni$ end])
- (f$)calcHeterozygosity(o<Genome> genomes, [No<Mut> muts], [Ni$ start], [Ni$ end])
- (f$)calcInbreedingLoad(o<Genome> genomes, [No<MutType> mutType])
- (f$)calcPairHeterozygosity(o<Genome>$ genome1, o<Genome>$ genome2, [Ni$ start], [Ni$ end],
  [l$ infiniteSites])
- (f$)calcVA(o<Ind> individuals, io<MutType>$ mutType)
- (f$)calcWattersonsTheta(o<Genome> genomes, [No<Mut> muts], [Ni$ start], [Ni$ end])
```

## SLiM nucleotide-based utilities:

```
- (s)codonsToAminoAcids(i codons, [li$ long], [l$ paste]) (nuc)
- (is)codonsToNucleotides(i codons, [s$ format]) (nuc)
- (f)mm16To256(f mutationMatrix) (nuc)
- (f)mmJukesCantor(f$ alpha) (nuc)
- (f)mmKimura(f$ alpha, f$ beta) (nuc)
- (i)nucleotideCounts(is sequence) (nuc)
- (f)nucleotideFrequencies(is sequence) (nuc)
- (i)nucleotidesToCodons(is sequence) (nuc)
- (is)randomNucleotides(i$ length, [Nif basis], [s$ format]) (nuc)
```

## SLiM other utilities:

```
- (f)summarizeIndividuals(o<Ind> individuals, i dim, n spatialBounds, s$ operation,
  [Nlif$ empty], [l$ perUnitArea], [Ns$ spatiality])
- (o<Dictionary>)treeSeqMetadata(s$ filePath, [l$ userData])
```

## Community:

*Superclass: Dictionary*

```
allGenomicElementTypes => (o<GEType>)
allInteractionTypes => (o<IntType>)
allMutationTypes => (o<MutType>)
allScriptBlocks => (o<SEBlock>)
allSpecies => (o<Species>)
allSubpopulations => (o<Subpop>)
cycleStage <=> (s$)
logFiles => (o<LogFile>)
modelType => (s$)
tag <=> (i$)
tick <=> (i$)
verbosity <=> (i$)
- (o<LogFile>$.createLogFile(s$ filePath, [Ns initialContents], [l$ append], [l$ compress],
  [s$ sep], [Ni$ logInterval], [Ni$ flushInterval]))
- (void)deregisterScriptBlock(io<SEBlock> scriptBlocks)
- (o<GEType>)genomicElementTypesWithIDs(i ids)
- (o<IntType>)interactionTypesWithIDs(i ids)
- (o<MutType>)mutationTypesWithIDs(i ids)
- (void)outputUsage(void)
- (o<SEBlock>$.register[First/Early/Late]Event(Nis$ id, s$ source, [Ni$ start], [Ni$ end]))
- (o<SEBlock>$.registerInteractionCallback(Nis$ id, s$ source, io<IntType>$ intType,
  [Nio<Subpop>$ subpop], [Ni$ start], [Ni$ end]))
- (o<SEBlock>$.rescheduleScriptBlock(io<SEBlock>$ block, [Ni$ start], [Ni$ end],
  [Ni ticks]))
- (o<SEBlock>)scriptBlocksWithIDs(i ids)
- (void)simulationFinished(void)
- (o<Species>)speciesWithIDs(i ids)
- (o<Subpop>)subpopulationsWithIDs(i ids)
```