

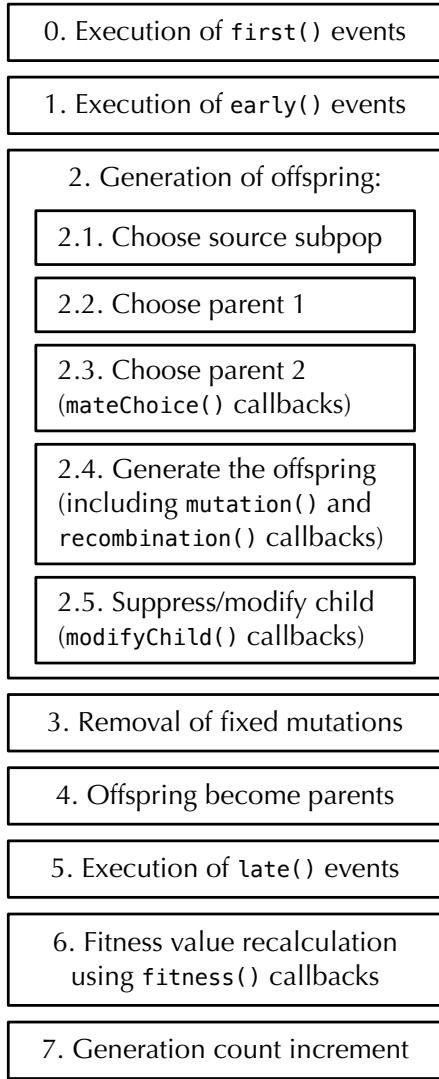
```
slim -version | -usage | -help | -testEidos | -testSLiM |
[-seed <seed>] [-time] [-mem] [-Memhist] [-long [<l>]] [-x] [-define <def>] <script file>
```

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

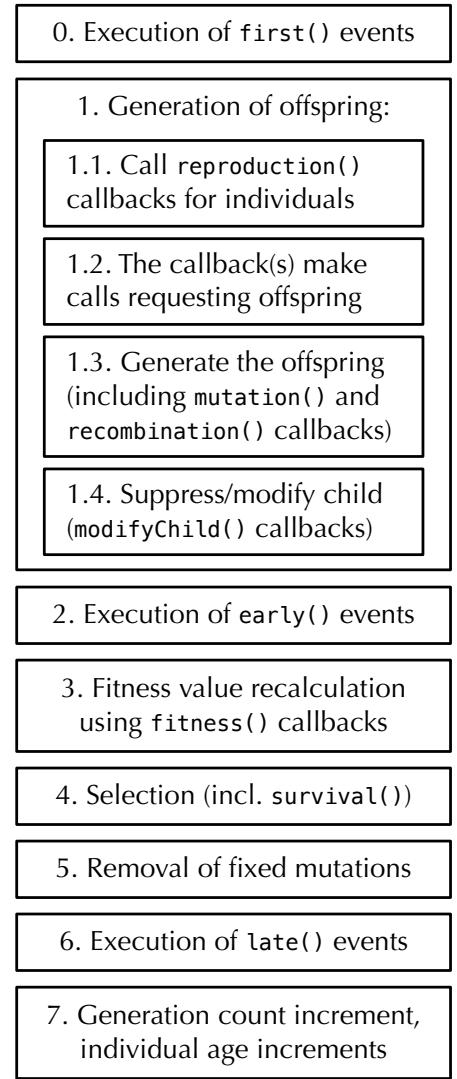
```
(ret-type) functionName(params) { ... }
[<id>] initialize() { ... }
[<id>] [gen1 [: gen2]] [first() | early() | late()] { ... }
[<id>] [gen1 [: gen2]] fitness(<mutTypeId> [, <subpopId>]) { ... }
[<id>] [gen1 [: gen2]] interaction(<intTypeId> [, <subpopId>]) { ... }
[<id>] [gen1 [: gen2]] mateChoice([<subpopId>]) { ... } (WF)
[<id>] [gen1 [: gen2]] modifyChild([<subpopId>]) { ... }
[<id>] [gen1 [: gen2]] recombination([<subpopId>]) { ... }
[<id>] [gen1 [: gen2]] mutation([<mutTypeId> [, <subpopId>]]) { ... }
[<id>] [gen1 [: gen2]] reproduction([<subpopId> [, <sex>]]) ... (nonWF)
[<id>] [gen1 [: gen2]] survival([<subpopId>]) ... (nonWF)
```

user-defined function
initialize() callback
events (default early())
fitness() callback
interaction() callback
mateChoice() callback
modifyChild() callback
recombination() callback
mutation() callback
reproduction() callback
survival() callback

The sequence of events within one generation in WF models.



The sequence of events within one generation in nonWF models.



* : generation 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<GEElement>)initializeGenomicElement(io<GEType> genomicElementType, i start, i end)
(o<GEType>$)initializeGenomicElementType(is$ 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$ distributionType, ...) (nuc)
(void)initializeRecombinationRate(n rates, [Ni ends], [s$ sex])
(void)initializeSex(s$ chromosomeType, [n$ xDominanceCoeff])
(void)initializeSLiMModelType(s$ modelType)
(void)initializeSLiMOptions([l$ keepPedigrees], [s$ dimensionality], [s$ periodicity],
    [i$ mutationRuns], [l$ preventIncidentalSelfing], [l$ nucleotideBased])
(void)initializeTreeSeq([l$ recordMutations], [Nif$ simplificationRatio],
    [Ni$ simplificationInterval], [l$ checkCoalescence], [l$ runCrosschecks],
    [l$ retainCoalescentOnly], [Ns$ timeUnit])
```

SLiM callbacks:

fitness():
 mut (o<Mutation>\$)
 homozygous (l\$)
 relFitness (f\$)
 individual (o<Ind>\$)
 genome1 (o<Genome>\$)
 genome2 (o<Genome>\$)
 subpop (o<Subpop>\$)

float\$: fitness effect

reproduction(): (nonWF)
 individual (o<Ind>\$)
 genome1 (o<Genome>\$)
 genome2 (o<Genome>\$)
 subpop (o<Subpop>\$)

void: (no return)

mutation():
 mut (o<Mutation>\$)
 genome (o<Genome>\$)
 element (o<GEElement>\$)
 originalNuc (i\$)
 parent (o<Ind>\$)
 subpop (o<Subpop>\$)

T: accept the mutation

F: reject the mutation

o<Mut>\$: use a substitute

SLiMEidosBlock (SEBlock):

Superclass: Dictionary

```
active <-> (i$)
end => (i$)
id => (i$)
source => (s$)
start => (i$)
tag <-> (i$)
type => (s$)
```

modifyChild():
 child (o<Ind>\$)
 childGenome1 (o<Genome>\$)
 childGenome2 (o<Genome>\$)
 childIsFemale (l\$)
 parent1 (o<Ind>\$)
 parent1Genome1 (o<Genome>\$)
 parent1Genome2 (o<Genome>\$)
 isCloning (l\$)
 isSelfing (l\$)
 parent2 (o<Ind>\$)
 parent2Genome1 (o<Genome>\$)
 parent2Genome2 (o<Genome>\$)
 subpop (o<Subpop>\$)
 sourceSubpop (o<Subpop>\$)

T: accept the proposed child
 F: reject the proposed child

recombination():
 individual (o<Ind>\$)
 genome1 (o<Genome>\$)
 genome2 (o<Genome>\$)
 subpop (o<Subpop>\$)
 breakpoints (i)
~~geStarts (i)~~
~~geEnds (i)~~

T: breakpoints changed
 F: breakpoints unchanged

mateChoice(): (WF)
 individual (o<Ind>\$)
 genome1 (o<Genome>\$)
 genome2 (o<Genome>\$)
 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

interaction():
 distance (f\$)
 strength (f\$)
 exerter (o<Ind>\$)
 receiver (o<Ind>\$)
 subpop (o<Subpop>\$)

float\$: interaction strength

survival(): (nonWF)
 individual (o<Ind>\$)
 subpop (o<Subpop>\$)
 surviving (l\$)
 fitness (f\$)
 draw (f\$)

T: individual survives

F: individual dies

NULL: SLiM's decision

o<Subpop>\$: move to subpop

SLiM globals:

```
sim (o<SLiMSim>$)
g1, ... (o<GEType>$)
i1, ... (o<IntType>$)
m1, ... (o<MutType>$)
p1, ... (o<Subpop>$)
s1, ... (o<SEBlock>$)
self (o<SEBlock>$)
```

SLiMgui quick help:
 opt-click on keyword

Code completion:
 escape (esc)

SLiMSim:

Superclass: *Dictionary*

```

chromosome => (o<Chromosome>$)
chromosomeType => ($$)
dimensionality => ($$)
dominanceCoeffX <-> (f$)
generation <-> (i$)
generationStage <-> ($$)
genomicElementTypes => (o<GEType>)
interactionTypes => (o<IntType>)
logFiles => (o<LogFile>)
modelType => ($$)
mutationTypes => (o<MutType>)
mutations => (o<Mut>)
nucleotideBased => (l$) (nuc)
periodicity => (string$)
scriptBlocks => (o<SEBlock>)
sexEnabled => (l$)
subpopulations => (o<Subpop>)
substitutions => (o<Substitution>)
tag <-> (i$)
verbosity <-> (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<LogFile>$)createLogFile($ filePath, [Ns initialContents], [l$ append], [l$ compress],
    [$ sep], [Ni$ logInterval], [Ni$ flushInterval])
- (void)deregisterScriptBlock(io<SEBlock> scriptBlocks)
- (i$)individualsWithPedigreeIDs(i pedigreeIDs, [Nio<Subpop> subpops])
- (i$)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])
- (void)outputUsage(void)
- (i$)readFromPopulationFile($ filePath)
- (void)recalculateFitness([Ni$ generation])
- (o<SEBlock>$)register[First/Early/Late]Event(Nis$ id, $ source, [Ni$ start], [Ni$ end])
- (o<SEBlock>$)registerFitnessCallback(Nis$ id, $ source, Nio<MutType>$ mutType,
    [Nio<Subpop>$ subpop], [Ni$ start], [Ni$ end])
- (o<SEBlock>$)registerInteractionCallback(Nis$ id, $ source, io<IntType>$ intType,
    [Nio<Subpop>$ subpop], [Ni$ start], [Ni$ end])
- (o<SEBlock>$)register[MateChoice (WF)/ModifyChild/Recombination]Callback(Nis$ id, $ source,
    [Nio<Subpop>$ subpop], [Ni$ start], [Ni$ end])
- (o<SEBlock>$)registerMutationCallback(Nis$ id, $ source, [Nio<MutType>$ mutType],
    [Nio<Subpop>$ subpop], [Ni$ start], [Ni$ end])
- (o<SEBlock>$)registerReproductionCallback(Nis$ id, $ source, [Nio<Subpop>$ subpop],
    [Ns$ sex], [Ni$ start], [Ni$ end]) (nonWF)
- (o<SEBlock>$)rescheduleScriptBlock(io<SEBlock>$ block, [Ni$ start], [Ni$ end],
    [Ni generations])
- (void)simulationFinished(void)
- (o<Mut>)subsetMutations([No<Mut>$ exclude], [Nio<MutType>$ mutType], [Ni$ position],
    [Nis$ nucleotide], [Nl$ tag], [Nl$ id])
- (l$)treeSeqCoalesced(void)
- (void)treeSeqOutput($ path, [l$ simplify], [l$ includeModel], [No$ metadata])
- (void)treeSeqRememberIndividuals(o<Ind> individuals, [l$ permanent])
- (void)treeSeqSimplify(void)

```

Fitness effects of mutations:

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

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

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$) (WF)
spatialBounds => (f)
tag <-> (i$)

- (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)spatialMapView(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)
```

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)
```

Mutation (Mut):

Superclass: *Dictionary*

```
id => (i$)
isFixed => (l$)
isSegregating => (l$)
mutationType => (o<MutType>$)
nucleotide <-> (s$) (nuc)
nucleotideValue <-> (i$) (nuc)
originGeneration => (i$)
position => (i$)
selectionCoeff => (f$)
subpopID <-> (i$)
tag <-> (i$)

- (void)setMutationType(io<MutType>$ mutType)
- (void)setSelectionCoeff(f$ selectionCoeff)
```

Substitution:

Superclass: *Dictionary*

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

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$)
recombinationEndPositions[F|M] => (i)
recombinationRates[F|M] => (f)
tag <-> (i$)

- (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])
```

MutationType (MutType):

Superclass: *Dictionary*

```
color <-> (s$)
colorSubstitution <-> (s$)
convertToSubstitution <-> (l$)
distributionParams => (f)
distributionType => (s$)
dominanceCoeff <-> (f$)
id => (i$)
mutationStackGroup <-> (i$)
mutationStackPolicy <-> (s$)
nucleotideBased => (l$) (nuc)
tag <-> (i$)

- (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, [Ni originGeneration],
    [Nio<Subpop> originSubpop], [Nis nucleotide])
+ (o<Mut>)addNewMutation(io<MutType> mutationType, n selectionCoeff, i position,
    [Ni originGeneration], [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$)positionsOfType(io<MutType>$ 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$)clippedIntegral(o<Ind> individuals)
- (f$)distance(o<Ind> individuals1, [No<Ind> individuals2])
- (f$)distanceToPoint(o<Ind> individuals1, f point)
- (o<Ind>)drawByStrength(o<Ind>$ individual, [i$ count])
- (void)evaluate([Nio<Subpop> subpops], [l$ immediate])
- (i$)interactingNeighborCount(o<Ind> individuals)
- (f$)interactionDistance(o<Ind> individuals1, [No<Ind> individuals2])
- (f$)localPopulationDensity(o<Ind> individuals)
- (o<Ind>)nearestInteractingNeighbors(o<Ind>$ individual, [i$ count])
- (o<Ind>)nearestNeighbors(o<Ind>$ individual, [i$ count])
- (o<Ind>)nearestNeighborsOfPoint(io<Subpop>$ subpop, f point, [i$ count])
- (void)setInteractionFunction(s$ functionType, ...)
- (f$)strength(o<Ind>$ receiver, [No<Ind> exerter])
- (f$)totalOfNeighborStrengths(o<Ind> individuals)
- (void)unevaluate(void)
```

GenomicElement (GElement):

Superclass: *Object*

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

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

GenomicElementType (GEType):

Superclass: Dictionary

```

color <-> ($$)
id => (i$)
mutationFractions => (f)
mutationMatrix => (f) (nuc)
mutationTypes => (o<MutType>)
tag <-> (i$)

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

```

LogFile:

Superclass: Dictionary

```

filePath => ($$)
logInterval => (i$)
precision <-> (i$)
tag <-> (i$)

- (void)addCustomColumn($ columnName, $ source, [* context])
- (void)addGeneration(void)
- (void)addGenerationStage(void)
- (void)addMeanSDColumns($ columnName, $ source, [* context])
- (void)addPopulationSexRatio(void)
- (void)addPopulationSize(void)
- (void)addSubpopulationSexRatio(io<Subpop>$ subpop)
- (void)addSubpopulationSize(io<Subpop>$ subpop)
- (void)flush(void)
- (void)logRow(void)
- (void)setLogInterval([Ni$ logInterval])
- (void)setFilePath($ filePath, [Ns initialContents], [l$ append], [Nl$ compress], [Ns$ sep])

```

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$)calcPairHeterozygosity(o<Genome>$ 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, [lis 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])

```

SLiMgui: (in SLiMgui only)

Superclass: Dictionary

```

pid => (i$)

- (void)openDocument($ filePath)
- (void)pauseExecution(void)

```