**BASE PAPER TITLE:** HEART DISEASE IDENTIFICATION METHOD USING MACHINE LEARNING CLASSIFICATION IN E-HEALTHCARE

**OWN TITLE:**

**ABSTRACT:**

Heart illness is one of the intricate infections and around the world numerous individuals experienced this sickness. On schedule and proficient recognizable proof of heart illness assumes a critical function in medical care, especially in the field of cardiology. In this article, we proposed a productive and exact framework to determination heart sickness and the framework depends on AI procedures. The framework is created dependent on order calculations incorporates Support vector machine, Logistic relapse, Artificial neural organization, K-closest neighbor, Naïve bayous, and Decision tree while standard highlights choice calculations have been utilized, for example, Relief, Minimal repetition maximal pertinence, Least supreme shrinkage determination administrator furthermore, Local learning for eliminating unimportant and excess highlights. We likewise proposed novel quick restrictive common data highlight choice calculation to tackle include determination issue. The highlights determination calculations are utilized for highlights determination to expand the order exactness and lessen the execution time of characterization framework. Moreover, the leave one subject out cross-approval technique has been utilized for learning the accepted procedures of model appraisal and for hyper parameter tuning. The presentation estimating measurements are utilized for appraisal of the exhibitions of the classifiers. The exhibitions of the classifiers have been kept an eye on the chose highlights as chosen by highlights determination calculations. The test results show that the proposed include determination calculation (FCMIM) is plausible with classifier uphold vector machine for planning a significant level smart framework to recognize heart illness. The proposed finding framework (FCMIM-SVM) accomplished great precision when contrasted with recently proposed techniques. Furthermore, the proposed framework can without much of a stretch be actualized in medical care for the distinguishing proof of heart illness.

**SCOPE OF PROJECT:**

Fast conditional mutual information (FCMIM) FS calculation for includes determination and afterward these highlights are contribution to classifiers for improving expectation precision and lessening calculation time. The classifiers exhibitions have been analyzed on highlights chose by the standard state of the workmanship FS calculations with the chose highlights of the proposed FS calculation. Recognize powerless highlights from the dataset which influence the exhibition of the classifiers. Finally heart illness recognizable proof framework (FCMIM-SVM) viably distinguishes the HD (Heart Disease).

**OBJECTIVES:**

Machine learning based diagnosis method for the identification of HD in this research work. Machine learning predictive models include ANN, LR, K-NN, SVM, DT, and NB is used for the identification of HD. The standard state of the art features selection algorithms, such as Relief, mRMR, LASSO and Local-learning-based features-selection (LLBFS) have been used to select the features. We also proposed fast conditional mutual information (FCMIM) features selection algorithm for features selection. Leave-one-subject-out cross-validation (LOSO) technique has been applied to select the best hyper-parameters for best model selection.

**EXISTING SYSTEM:**

Expert choice system in light of AI classifiers and the use of fake fluffy rationale is successfully finding the HD therefore, the proportion of death diminishes and The Cleveland heart illness informational index was utilized by different analysts also for the distinguishing proof issue of HD. The machine learning prescient models need appropriate information for preparing and testing. The presentation of AI model can be expanded whenever adjusted dataset is use for preparing and testing of the model. Moreover, the model prescient abilities can improve by utilizing appropriate and related highlights from the information. Hence, information adjusting and highlight determination is altogether significant for model execution improvement. In writing different analysis strategies have been proposed by different analysts, anyway these strategies are most certainly not successfully analysis HD.

**DIS-ADVANTAGES:**

* Lack of prediction accuracy.
* High computation time for prediction of HD.
* There are no successfully hybrid algorithms.

**PROPOSED SYSTEM:**

We proposed an AI based conclusion technique for the ID of HD in this examination work. Machine learning prescient models incorporate ANN, LR, K-NN, SVM, DT, and NB are utilized for the recognizable proof of HD. The standard best in class highlights determination calculations, for example, Relief, mRMR, LASSO and Local-learning-based features-selection (LLBFS) have been utilized to choose the highlights. We additionally proposed fast conditional mutual information (FCMIM) highlights choice calculation for highlights determination. Leave-one-subject-out cross-validation (LOSO) procedure has been applied to choose the best hyper-boundaries for best model choice. Aside from this, diverse presentation appraisal measurements have been utilized for classifiers exhibitions assessment. The proposed strategy has been tried on Cleveland HD dataset. Moreover, the exhibition of the proposed method have been analyzed with best in class existing strategies in the writing, for example, NB , Three stage ANN (Artificial neural Network) analysis framework, Neural network ensembles (NNE) , ANN-Fuzzy-AHP diagnosis system (AFP), Adaptive-weighted-Fuzzy-system-ensemble (AWFSE) . The examination study has the accompanying commitments.

**ADVANTAGES:**

* High accuracy on diagnosis of HD.
* Using hybrid algorithms with new FCMIM-SVM algorithm.
* Reducing computation time.

**SYSTEM CONFIGURATION**

# H/W SYSTEM CONFIGURATION:

# Processor - Intel

* Speed - 1.1 GHz
* RAM - 4 Gb
* Hard Disk - 260 GB

# S/W SYSTEM CONFIGURATION:

* Operating System - Windows 7/8/10
* Front End - Html/ Css
* Scripts - R language
* Tool - RStudio v1.3.1093

**Problem Identification**

Expert choice system in light of AI classifiers and the use of fake fluffy rationale is successfully finding the HD therefore, the proportion of death diminishes and The Cleveland heart illness informational index was utilized by different analysts also for the distinguishing proof issue of HD. The machine learning prescient models need appropriate information for preparing and testing. The presentation of AI model can be expanded whenever adjusted dataset is use for preparing and testing of the model. Moreover, the model prescient abilities can improve by utilizing appropriate and related highlights from the information. Hence, information adjusting and highlight determination is altogether significant for model execution improvement. In writing different analysis strategies have been proposed by different analysts, anyway these strategies are most certainly not successfully analysis HD.

**Modules:**

* DATA PRE-PROCESSING
* FEATURE SELECTION AND REDUCTION
* CLASSIFICATION MODELLING
* PERFORMANCE MEASURES
* PREDICTION USING CNN ALGORITHM

**DATA PRE-PROCESSING**

* Heart disease data is pre-processed after collection of various records. The dataset contains a total of patient records, where records are with some missing values. Those records have been removed from the dataset and the remaining patient records are used in pre-processing. The multiclass variable and binary classification are introduced for the attributes of the given dataset. The multi-class variable is used to check the presence or absence of heart disease. In the instance of the patient having heart disease, the value is set to else the value is set to indicating the absence of heart disease in the patient. The pre-processing of data is carried out by converting medical records into diagnosis values. The results of data pre-processing for patient records indicate that records show the value of establishing the presence of heart disease while the remaining reflected the value of 0 indicating the absence of heart disease.

**FEATURE SELECTION AND REDUCTION**

From among the attributes of the data set, two attributes pertaining to age and sex are used to identify the personal information of the patient. The remaining attributes are considered important as they contain vital clinical records. Clinical records are vital to diagnosis and learning the severity of heart disease. As previously mentioned in this experiment, convolutional neural network used. The experiment was repeated with all the ML techniques using all 13 attribute.

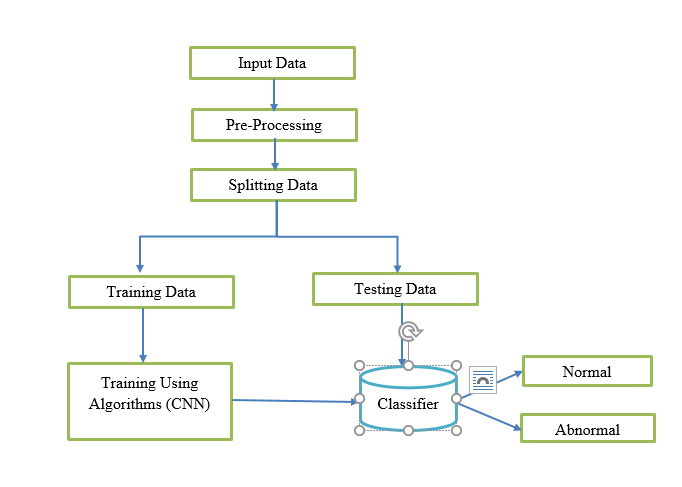
**CLASSIFICATION MODELLING**

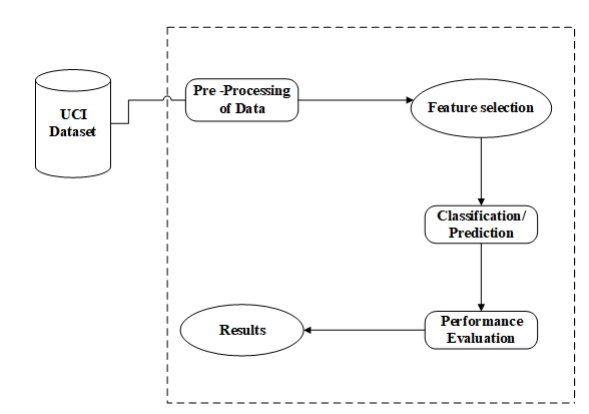
The clustering of datasets is done on the basis of the variables and criteria of Decision Tree (DT) features. Then, the classifiers are applied to each clustered dataset in order to estimate its performance. The best performing models are identified from the above results based on their low rate of error. The performance is further optimized by choosing the DT cluster with a high rate of error and extraction of its corresponding classifier features. The performance of the classifier is evaluated for error optimization on this data set.

**DISCUSSION OF HRFLM TO IMPROVE THE RESULTS**

The UCI dataset is further classified into 8 types of datasets based on classification rules. The classification rules Each dataset is further classified and processed by R Studio Rattle. The results are generated by applying the classification rule for the dataset. The classification rules generated based on the rule after data pre-processing is done. After pre-processing, the data’s three best ML techniques are chosen and the results are generated. The various datasets with DT, RF, LM are applied to find out the best classification method. The results show that RF and LM are the best. The RF error rate for dataset 4 is high (20.9%) compared to the other datasets. The LM method for the dataset is the best (9.1%) compared to DT and RF methods. We propose convolutional neural network method to improve the results. results of the proposed method

**Block Diagram:**

****

****

**LITERATURE SURVEY**

Title: IDENTIFYING THE PREDICTIVE CAPABILITY OF MACHINE LEARNING CLASSIFIERS FOR DESIGNING HEART DISEASE DETECTION SYSTEM.

Author: AMIN UL HAQ1 , JIANPING LI .

Year:2019

Abstract:

The analysis of heart illnesses through receipt based procedures just as common clinical based techniques are most certainly not dependable. On other hand, non-receipt based procedures are more compelling for coronary illness determination. Along these lines, we check the capacity of different Machine Learning (ML) classifiers and profound learning classifier for coronary illness distinguishing proof in this paper. Six AI classifiers and BPNN were utilized in request to check which one classifier is more powerful for analysis the coronary illness. The element choice calculation Help was utilized for determination of significant highlights and on these chosen highlights, classifiers exhibitions were likewise registered. Troupe AI methods (boosting, sacking, stacking) were utilized to additional expansion the classifiers execution. Besides, cross-approval methods k-folds was additionally utilized. Moreover in reverse engendering neural network (BPNN) was additionally utilized for characterization reason since profound learning calculation not need highlight determination calculations and it naturally select significant highlights for great outcome. In view of model execution assessment measurements the SVM (RBF) performed brilliantly on full highlights accomplished exactness 86%, and 88% precision on chose includes as analyzed different classifiers. Through Ensemble learning methods, SVM acquired the grouping precision 92.30%.

**Paper 2:**

Title: Decision Support System for Choosing Daycare in Surabaya City Using Analytical Hierarchy Process (AHP).

Author: Kholid Fathoni, Ira Prasetyaningrum

Year:2019

Abstract: Financial issues prompted changes in ladies' jobs where ladies are deciding to work, so they can't continuously be with the youngsters and will in general pick kid care in childcare when they are functioning as an answer. The schedules make them come up short on any data about childcares particularly profiles and areas. So guardians need a data framework that can give data about childcare and exhort the choice. The motivation behind this examination is to give a framework that gives data of every childcare furthermore, childcare suggestions, so it tends to be utilized by guardians. In this exploration, dynamic framework was led utilizing Analytical Hierarchy Process (AHP), which is a multi-standards dynamic strategy. A proposal is made by client's position. The finish of this exploration is that the framework can give data and suggestions about childcare and capable to address the issues of guardians.

**Paper3:**

Title: Predicting Heart Disease at Early Stages using Machine Learning: A Survey

Author: Rahul Katarya.

Year:2018

Abstract: Anticipating and identification of coronary illness has continuously been a basic and testing task for medical care specialists. Medical clinics and different centers are offering costly treatments and activities to treat heart infections. Along these lines, foreseeing coronary illness at the beginning phases will be helpful to individuals around the globe with the goal that they will make vital moves previously getting extreme. Coronary illness is a huge issue in later times; the primary explanation behind this sickness is the admission of liquor, tobacco, and absence of actual exercise. Throughout the long term, machine learning shows powerful outcomes in settling on choices and forecasts from the expansive arrangement of information delivered by the wellbeing care industry. A portion of the regulated AI methods utilized in this expectation of coronary illness are fake neural organization (ANN), choice tree (DT), arbitrary woods (RF), uphold vector machine (SVM), innocent Bayes) (NB) and knearest neighbor calculation. Moreover, the exhibitions of these calculations are summed up.

**Paper4:**

Title: Radial basis function Neural Network for Prediction of Cardiac Arrhythmias based on Heart rate time series

Author: J. P. Kelwade Dr. S. S. Salankar

Year:2019

Abstract: Anticipating and identification of coronary illness has continuously been a basic and testing task for medical care specialists. Medical clinics and different centers are offering costly treatments and activities to treat heart infections. Along these lines, foreseeing coronary illness at the beginning phases will be helpful to individuals around the globe with the goal that they will make vital moves previously getting extreme. Coronary illness is a huge issue in later times; the primary explanation behind this sickness is the admission of liquor, tobacco, and absence of actual exercise. Throughout the long term, machine learning shows powerful outcomes in settling on choices and forecasts from the expansive arrangement of information delivered by the wellbeing care industry. A portion of the regulated AI methods utilized in this expectation of coronary illness are fake neural organization (ANN), choice tree (DT), arbitrary woods (RF), uphold vector machine (SVM), innocent Bayes) (NB) and knearest neighbor calculation. Moreover, the exhibitions of these calculations are summed up.

# PYTHON INTRODUCTION:

**Python** is an [interpreted](https://en.wikipedia.org/wiki/Interpreted_language), [high-level](https://en.wikipedia.org/wiki/High-level_programming_language) and [general-purpose programming language](https://en.wikipedia.org/wiki/General-purpose_programming_language). Python's design philosophy emphasizes [code readability](https://en.wikipedia.org/wiki/Code_readability) with its notable use of [significant indentation](https://en.wikipedia.org/wiki/Off-side_rule). Its [language constructs](https://en.wikipedia.org/wiki/Language_construct) and [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming) approach aim to help [programmers](https://en.wikipedia.org/wiki/Programmers) write clear, logical code for small and large-scale projects.

Python is [dynamically-typed](https://en.wikipedia.org/wiki/Dynamic_programming_language) and [garbage-collected](https://en.wikipedia.org/wiki/Garbage_collection_(computer_science)). It supports multiple [programming paradigms](https://en.wikipedia.org/wiki/Programming_paradigms), including [structured](https://en.wikipedia.org/wiki/Structured_programming) (particularly, [procedural](https://en.wikipedia.org/wiki/Procedural_programming)), [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming) and [functional programming](https://en.wikipedia.org/wiki/Functional_programming). Python is often described as a "batteries included" language due to its comprehensive [standard library](https://en.wikipedia.org/wiki/Standard_library).

[Guido van Rossum](https://en.wikipedia.org/wiki/Guido_van_Rossum) began working on Python in the late 1980s, as a successor to the [ABC programming language](https://en.wikipedia.org/wiki/ABC_(programming_language)), and first released it in 1991 as Python 0.9.0. Python 2.0 was released in 2000 and introduced new features, such as [list comprehensions](https://en.wikipedia.org/wiki/List_comprehension) and a garbage collection system using [reference counting](https://en.wikipedia.org/wiki/Reference_counting) and was discontinued with version 2.7.18 in 2020. Python 3.0 was released in 2008 and was a major revision of the language that is not completely [backward-compatible](https://en.wikipedia.org/wiki/Backward_compatibility) and much Python 2 code does not run unmodified on Python 3.

## PYTHON HISTORY

Python was conceived in the late 1980s by [Guido van Rossum](https://en.wikipedia.org/wiki/Guido_van_Rossum) at [Centrum Wiskunde & Informatica](https://en.wikipedia.org/wiki/Centrum_Wiskunde_%26_Informatica) (CWI) in the [Netherlands](https://en.wikipedia.org/wiki/Netherlands) as a successor to [ABC programming language](https://en.wikipedia.org/wiki/ABC_(programming_language)), which was inspired by [SETL](https://en.wikipedia.org/wiki/SETL), capable of [exception handling](https://en.wikipedia.org/wiki/Exception_handling) and interfacing with the [Amoeba](https://en.wikipedia.org/wiki/Amoeba_(operating_system)) operating system.[[10]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-faq-created-10) Its implementation began in December 1989. Van Rossum shouldered sole responsibility for the project, as the lead developer, until 12 July 2018, when he announced his "permanent vacation" from his responsibilities as Python's [Benevolent Dictator for Life](https://en.wikipedia.org/wiki/Benevolent_Dictator_For_Life), a title the Python community bestowed upon him to reflect his long-term commitment as the project's chief decision-maker. He now shares his leadership as a member of a five-person steering council. In January 2019, active Python core developers elected Brett Cannon, Nick Coghlan, Barry Warsaw, Carol Willing and Van Rossum to a five-member "Steering Council" to lead the project. Guido van Rossum has since then withdrawn his nomination for the 2020 Steering council.

Python 2.0 was released on 16 October 2000, with many major new features, including a [cycle-detecting](https://en.wikipedia.org/wiki/Cycle_detection) [garbage collector](https://en.wikipedia.org/wiki/Garbage_collection_(computer_science)) and support for [Unicode](https://en.wikipedia.org/wiki/Unicode).[[48]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-newin-2.0-48)

Python 3.0 was released on 3 December 2008. It was a major revision of the language that is not completely [backward-compatible](https://en.wikipedia.org/wiki/Backward_compatibility). Many of its major features were [backported](https://en.wikipedia.org/wiki/Backporting) to Python 2.6.x and 2.7.x version series. Releases of Python 3 include the 2to3 utility, which automates (at least partially) the translation of Python 2 code to Python 3.

Python 2.7's [end-of-life](https://en.wikipedia.org/wiki/End-of-life_(product)) date was initially set at 2015 then postponed to 2020 out of concern that a large body of existing code could not easily be forward-ported to Python 3. No more security patches or other improvements will be released for it. With Python 2's [end-of-life](https://en.wikipedia.org/wiki/End-of-life_(product)), only Python 3.6.x and later are supported.

Python 3.9.2 and 3.8.8 were expedited as all versions of Python (including 2.7) had security issues, leading to possible [remote code execution](https://en.wikipedia.org/wiki/Remote_code_execution) and [web cache poisoning](https://en.wikipedia.org/wiki/Cache_poisoning).

## DESIGN PHILOSOPHY AND FEATURES

Python is a [multi-paradigm programming language](https://en.wikipedia.org/wiki/Multi-paradigm_programming_language). [Object-oriented programming](https://en.wikipedia.org/wiki/Object-oriented_programming) and [structured programming](https://en.wikipedia.org/wiki/Structured_programming) are fully supported, and many of its features support [functional programming](https://en.wikipedia.org/wiki/Functional_programming) and [aspect-oriented programming](https://en.wikipedia.org/wiki/Aspect-oriented_programming) (including by [metaprogramming](https://en.wikipedia.org/wiki/Metaprogramming) and [metaobjects](https://en.wikipedia.org/wiki/Metaobject) (magic methods)). Many other paradigms are supported via extensions, including [design by contract](https://en.wikipedia.org/wiki/Design_by_contract) and [logic programming](https://en.wikipedia.org/wiki/Logic_programming).

Python uses [dynamic typing](https://en.wikipedia.org/wiki/Dynamic_typing) and a combination of [reference counting](https://en.wikipedia.org/wiki/Reference_counting) and a cycle-detecting garbage collector for [memory management](https://en.wikipedia.org/wiki/Memory_management). It also features dynamic [name resolution](https://en.wikipedia.org/wiki/Name_resolution_(programming_languages)) ([late binding](https://en.wikipedia.org/wiki/Late_binding)), which binds method and variable names during program execution.

Python's design offers some support for [functional programming](https://en.wikipedia.org/wiki/Functional_programming) in the [Lisp](https://en.wikipedia.org/wiki/Lisp_(programming_language)) tradition. It has filter, map, and reduce functions; [list comprehensions](https://en.wikipedia.org/wiki/List_comprehension), [dictionaries](https://en.wikipedia.org/wiki/Associative_array), sets, and [generator](https://en.wikipedia.org/wiki/Generator_(computer_programming)) expressions. The standard library has two modules (itertools and functools) that implement functional tools borrowed from [Haskell](https://en.wikipedia.org/wiki/Haskell_(programming_language)) and [Standard ML](https://en.wikipedia.org/wiki/Standard_ML).

The language's core philosophy is summarized in the document The [Zen of Python](https://en.wikipedia.org/wiki/Zen_of_Python) (PEP 20), which includes [aphorisms](https://en.wikipedia.org/wiki/Aphorism) such as:

* Beautiful is better than ugly.
* Explicit is better than implicit.
* Simple is better than complex.
* Complex is better than complicated.
* Readability counts.

Rather than having all of its functionality built into its core, Python was designed to be highly [extensible](https://en.wikipedia.org/wiki/Extensibility) (with modules). This compact modularity has made it particularly popular as a means of adding programmable interfaces to existing applications. Van Rossum's vision of a small core language with a large standard library and easily extensible interpreter stemmed from his frustrations with [ABC](https://en.wikipedia.org/wiki/ABC_(programming_language)), which espoused the opposite approach.

Python strives for a simpler, less-cluttered syntax and grammar while giving developers a choice in their coding methodology. In contrast to [Perl](https://en.wikipedia.org/wiki/Perl)'s "[there is more than one way to do it](https://en.wikipedia.org/wiki/There_is_more_than_one_way_to_do_it)" motto, Python embraces a "there should be one— and preferably only one —obvious way to do it" design philosophy.[[69]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-PEP20-69) [Alex Martelli](https://en.wikipedia.org/wiki/Alex_Martelli), a [Fellow](https://en.wikipedia.org/wiki/Fellow) at the [Python Software Foundation](https://en.wikipedia.org/wiki/Python_Software_Foundation) and Python book author, writes that "To describe something as 'clever' is not considered a compliment in the Python culture."

Python's developers strive to avoid [premature optimization](https://en.wikipedia.org/wiki/Premature_optimization), and reject patches to non-critical parts of the [CPython](https://en.wikipedia.org/wiki/CPython) reference implementation that would offer marginal increases in speed at the cost of clarity.[[71]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-20-71) When speed is important, a Python programmer can move time-critical functions to extension modules written in languages such as C, or use [PyPy](https://en.wikipedia.org/wiki/PyPy), a [just-in-time compiler](https://en.wikipedia.org/wiki/Just-in-time_compilation). [Cython](https://en.wikipedia.org/wiki/Cython) is also available, which translates a Python script into C and makes direct C-level API calls into the Python interpreter.

An important goal of Python's developers is keeping it fun to use. This is reflected in the language's name—a tribute to the British comedy group [Monty Python](https://en.wikipedia.org/wiki/Monty_Python)—and in occasionally playful approaches to tutorials and reference materials, such as examples that refer to spam and eggs (from a [famous Monty Python sketch](https://en.wikipedia.org/wiki/Spam_(Monty_Python))) instead of the standard [foo and bar](https://en.wikipedia.org/wiki/Foobar).

A common [neologism](https://en.wikipedia.org/wiki/Neologism) in the Python community is pythonic, which can have a wide range of meanings related to program style. To say that code is pythonic is to say that it uses Python idioms well, that it is natural or shows fluency in the language, that it conforms with Python's minimalist philosophy and emphasis on readability. In contrast, code that is difficult to understand or reads like a rough transcription from another programming language is called unpythonic.

Users and admirers of Python, especially those considered knowledgeable or experienced, are often referred to as Pythonistas.

## SYNTAX AND SEMANTICS

Python is meant to be an easily readable language. Its formatting is visually uncluttered, and it often uses English keywords where other languages use punctuation. Unlike many other languages, it does not use [curly brackets](https://en.wikipedia.org/wiki/Curly_bracket_programming_language) to delimit blocks, and semicolons after statements are allowed but are rarely, if ever, used. It has fewer syntactic exceptions and special cases than [C](https://en.wikipedia.org/wiki/C_(programming_language)) or [Pascal](https://en.wikipedia.org/wiki/Pascal_(programming_language)).

### INDENTATION

Main article: [Python syntax and semantics § Indentation](https://en.wikipedia.org/wiki/Python_syntax_and_semantics#Indentation)

Python uses [whitespace](https://en.wikipedia.org/wiki/Whitespace_character) indentation, rather than [curly brackets](https://en.wikipedia.org/wiki/Curly_bracket_programming_language) or keywords, to delimit [blocks](https://en.wikipedia.org/wiki/Block_(programming)). An increase in indentation comes after certain statements; a decrease in indentation signifies the end of the current block. Thus, the program's visual structure accurately represents the program's semantic structure. This feature is sometimes termed the [off-side rule](https://en.wikipedia.org/wiki/Off-side_rule), which some other languages share, but in most languages indentation doesn't have any semantic meaning. The recommended indent size is four spaces.

### STATEMENTS AND CONTROL FLOW

Python's [statements](https://en.wikipedia.org/wiki/Statement_(computer_science)) include (among others):

* The [assignment](https://en.wikipedia.org/wiki/Assignment_(computer_science)) statement, using a single equals sign =.
* The [if](https://en.wikipedia.org/wiki/If-then-else) statement, which conditionally executes a block of code, along with else and elif (a contraction of else-if).
* The [for](https://en.wikipedia.org/wiki/Foreach#Python) statement, which iterates over an iterable object, capturing each element to a local variable for use by the attached block.
* The [while](https://en.wikipedia.org/wiki/While_loop#Python) statement, which executes a block of code as long as its condition is true.
* The [try](https://en.wikipedia.org/wiki/Exception_handling_syntax#Python) statement, which allows exceptions raised in its attached code block to be caught and handled by except clauses; it also ensures that clean-up code in a finally block will always be run regardless of how the block exits.
* The raise statement, used to raise a specified exception or re-raise a caught exception.
* The class statement, which executes a block of code and attaches its local namespace to a [class](https://en.wikipedia.org/wiki/Class_(computer_science)), for use in [object-oriented programming](https://en.wikipedia.org/wiki/Object-oriented_programming).
* The def statement, which defines a [function](https://en.wikipedia.org/wiki/Function_(computing)) or [method](https://en.wikipedia.org/wiki/Method_(computing)).
* The with statement, from Python 2.5 released in September 2006,[[82]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-82) which encloses a code block within a context manager (for example, acquiring a [lock](https://en.wikipedia.org/wiki/Lock_(computer_science)) before the block of code is run and releasing the lock afterwards, or opening a [file](https://en.wikipedia.org/wiki/Computer_file) and then closing it), allowing [resource-acquisition-is-initialization](https://en.wikipedia.org/wiki/Resource_acquisition_is_initialization) (RAII)-like behavior and replaces a common try/finally idiom.[[83]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-83)
* The [break](https://en.wikipedia.org/wiki/Break_statement) statement, exits from a loop.
* The continue statement, skips this iteration and continues with the next item.
* The del statement, removes a variable, which means the reference from the name to the value is deleted and trying to use that variable will cause an error. A deleted variable can be reassigned.
* The pass statement, which serves as a [NOP](https://en.wikipedia.org/wiki/NOP_(code)). It is syntactically needed to create an empty code block.
* The [assert](https://en.wikipedia.org/wiki/Assertion_(programming)) statement, used during debugging to check for conditions that should apply.
* The yield statement, which returns a value from a [generator](https://en.wikipedia.org/wiki/Generator_(computer_programming)#Python) function. From Python 2.5, yield is also an operator. This form is used to implement [coroutines](https://en.wikipedia.org/wiki/Coroutine).
* The return statement, used to return a value from a function.
* The import statement, which is used to import modules whose functions or variables can be used in the current program. There are three ways of using import: import <module name> [as <alias>] or from <module name> import \* or from <module name> import <definition 1> [as <alias 1>], <definition 2> [as <alias 2>]

The assignment statement (=) operates by binding a name as a [reference](https://en.wikipedia.org/wiki/Pointer_(computer_programming)) to a separate, dynamically-allocated [object](https://en.wikipedia.org/wiki/Object_(computer_science)). Variables may be subsequently rebound at any time to any object. In Python, a variable name is a generic reference holder and doesn't have a fixed [data type](https://en.wikipedia.org/wiki/Type_system) associated with it. However at a given time, a variable will refer to some object, which will have a type. This is referred to as [dynamic typing](https://en.wikipedia.org/wiki/Dynamic_type) and is contrasted with [statically-typed](https://en.wikipedia.org/wiki/Statically-typed) programming languages, where each variable may only contain values of a certain type.

Python does not support [tail call](https://en.wikipedia.org/wiki/Tail_call) optimization or [first-class continuations](https://en.wikipedia.org/wiki/First-class_continuations), and, according to Guido van Rossum, it never will.[[84]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-55-84)[[85]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-56-85) However, better support for [coroutine](https://en.wikipedia.org/wiki/Coroutine)-like functionality is provided in 2.5, by extending Python's [generators](https://en.wikipedia.org/wiki/Generator_(computer_programming)).[[86]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-57-86) Before 2.5, generators were [lazy](https://en.wikipedia.org/wiki/Lazy_evaluation) [iterators](https://en.wikipedia.org/wiki/Iterator); information was passed unidirectionally out of the generator. From Python 2.5, it is possible to pass information back into a generator function, and from Python 3.3, the information can be passed through multiple stack levels.[[87]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-58-87)

### EXPRESSIONS

Some Python [expressions](https://en.wikipedia.org/wiki/Expression_(computer_science)) are similar to those found in languages such as [C](https://en.wikipedia.org/wiki/C_(programming_language)) and [Java](https://en.wikipedia.org/wiki/Java_(programming_language)), while some are not:

* Addition, subtraction, and multiplication are the same, but the behavior of division differs. There are two types of divisions in Python. They are floor division (or integer division) // and floating-point/division.[[88]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-88) Python also uses the \*\* operator for exponentiation.
* From Python 3.5, the new @ infix operator was introduced. It is intended to be used by libraries such as [NumPy](https://en.wikipedia.org/wiki/NumPy) for [matrix multiplication](https://en.wikipedia.org/wiki/Matrix_multiplication).[[89]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-PEP465-89)[[90]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-Python3.5Changelog-90)
* From Python 3.8, the syntax :=, called the 'walrus operator' was introduced. It assigns values to variables as part of a larger expression.[[91]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-Python3.8Changelog-91)
* In Python, == compares by value, versus Java, which compares numerics by value[[92]](https://en.wikipedia.org/wiki/Python_(programming_language)" \l "cite_note-92) and objects by reference.[[93]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-93) (Value comparisons in Java on objects can be performed with the equals() method.) Python's is operator may be used to compare object identities (comparison by reference). In Python, comparisons may be chained, for example a <= b <= c.
* Python uses the words and, or, not for its boolean operators rather than the symbolic &&, ||, ! used in Java and C.
* Python has a type of expression termed a [list comprehension](https://en.wikipedia.org/wiki/List_comprehension#Python) as well as a more general expression termed a [generator](https://en.wikipedia.org/wiki/Generator_(computer_programming)) expression.[[67]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-59-67)
* [Anonymous functions](https://en.wikipedia.org/wiki/Anonymous_function) are implemented using [lambda expressions](https://en.wikipedia.org/wiki/Lambda_(programming)); however, these are limited in that the body can only be one expression.
* Conditional expressions in Python are written as x if c else y[[94]](https://en.wikipedia.org/wiki/Python_(programming_language)" \l "cite_note-AutoNT-60-94) (different in order of operands from the [c ? x : y](https://en.wikipedia.org/wiki/%3F:) operator common to many other languages).
* Python makes a distinction between [lists](https://en.wikipedia.org/wiki/List_(computer_science)) and [tuples](https://en.wikipedia.org/wiki/Tuple). Lists are written as [1, 2, 3], are mutable, and cannot be used as the keys of dictionaries (dictionary keys must be [immutable](https://en.wikipedia.org/wiki/Immutable) in Python). Tuples are written as (1, 2, 3), are immutable and thus can be used as the keys of dictionaries, provided all elements of the tuple are immutable. The + operator can be used to concatenate two tuples, which does not directly modify their contents, but rather produces a new tuple containing the elements of both provided tuples. Thus, given the variable t initially equal to (1, 2, 3), executing t = t + (4, 5) first evaluates t + (4, 5), which yields (1, 2, 3, 4, 5), which is then assigned back to t, thereby effectively "modifying the contents" of t, while conforming to the immutable nature of tuple objects. Parentheses are optional for tuples in unambiguous contexts.[[95]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-95)
* Python features sequence unpacking wherein multiple expressions, each evaluating to anything that can be assigned to (a variable, a writable property, etc.), are associated in an identical manner to that forming tuple literals and, as a whole, are put on the left-hand side of the equal sign in an assignment statement. The statement expects an iterable object on the right-hand side of the equal sign that produces the same number of values as the provided writable expressions when iterated through and will iterate through it, assigning each of the produced values to the corresponding expression on the left.[[96]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-96)
* Python has a "string format" operator %. This functions analogously to [printf](https://en.wikipedia.org/wiki/Printf_format_string) format strings in [C](https://en.wikipedia.org/wiki/C_(programming_language)), e.g. "spam=%s eggs=%d" % ("blah", 2) evaluates to "spam=blah eggs=2". In Python 3 and 2.6+, this was supplemented by the format() method of the str class, e.g. "spam={0} eggs={1}".format("blah", 2). Python 3.6 added "f-strings": blah = "blah"; eggs = 2; f'spam={blah} eggs={eggs}'.[[97]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-pep-0498-97)
* Strings in Python can be [concatenated](https://en.wikipedia.org/wiki/Concatenation), by "adding" them (same operator as for adding integers and floats). E.g. "spam" + "eggs" returns "spameggs". Even if your strings contain numbers, they are still added as strings rather than integers. E.g. "2" + "2" returns "22".
* Python has various kinds of [string literals](https://en.wikipedia.org/wiki/String_literal):
  + Strings delimited by single or double quote marks. Unlike in [Unix shells](https://en.wikipedia.org/wiki/Unix_shell), [Perl](https://en.wikipedia.org/wiki/Perl) and Perl-influenced languages, single quote marks and double quote marks function identically. Both kinds of string use the backslash (\) as an [escape character](https://en.wikipedia.org/wiki/Escape_character). [String interpolation](https://en.wikipedia.org/wiki/String_interpolation) became available in Python 3.6 as "formatted string literals".[[97]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-pep-0498-97)
  + Triple-quoted strings, which begin and end with a series of three single or double quote marks. They may span multiple lines and function like [here documents](https://en.wikipedia.org/wiki/Here_document) in shells, Perl and [Ruby](https://en.wikipedia.org/wiki/Ruby_(programming_language)).
  + [Raw string](https://en.wikipedia.org/wiki/Raw_string) varieties, denoted by prefixing the string literal with an r. Escape sequences are not interpreted; hence raw strings are useful where literal backslashes are common, such as [regular expressions](https://en.wikipedia.org/wiki/Regular_expression) and [Windows](https://en.wikipedia.org/wiki/Microsoft_Windows)-style paths. Compare "@-quoting" in [C#](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)).
* Python has [array index](https://en.wikipedia.org/wiki/Array_index) and [array slicing](https://en.wikipedia.org/wiki/Array_slicing) expressions on lists, denoted as a[key], a[start:stop] or a[start:stop:step]. Indexes are [zero-based](https://en.wikipedia.org/wiki/Zero-based), and negative indexes are relative to the end. Slices take elements from the start index up to, but not including, the stop index. The third slice parameter, called step or stride, allows elements to be skipped and reversed. Slice indexes may be omitted, for example a[:] returns a copy of the entire list. Each element of a slice is a [shallow copy](https://en.wikipedia.org/wiki/Shallow_copy).

In Python, a distinction between expressions and statements is rigidly enforced, in contrast to languages such as [Common Lisp](https://en.wikipedia.org/wiki/Common_Lisp), [Scheme](https://en.wikipedia.org/wiki/Scheme_(programming_language)), or [Ruby](https://en.wikipedia.org/wiki/Ruby_(programming_language)). This leads to duplicating some functionality. For example:

* [List comprehensions](https://en.wikipedia.org/wiki/List_comprehensions) vs. for-loops
* [Conditional](https://en.wikipedia.org/wiki/Conditional_(programming)) expressions vs. if blocks
* The eval() vs. exec() built-in functions (in Python 2, exec is a statement); the former is for expressions, the latter is for statements.

Statements cannot be a part of an expression, so list and other comprehensions or [lambda expressions](https://en.wikipedia.org/wiki/Lambda_(programming)), all being expressions, cannot contain statements. A particular case of this is that an assignment statement such as a = 1 cannot form part of the conditional expression of a conditional statement. This has the advantage of avoiding a classic C error of mistaking an assignment operator = for an equality operator == in conditions: if (c = 1) { ... } is syntactically valid (but probably unintended) C code but if c = 1: ... causes a syntax error in Python.

### METHODS

[Methods](https://en.wikipedia.org/wiki/Method_(programming)) on objects are [functions](https://en.wikipedia.org/wiki/Function_(programming)) attached to the object's class; the syntax instance.method(argument) is, for normal methods and functions, [syntactic sugar](https://en.wikipedia.org/wiki/Syntactic_sugar) for Class.method(instance, argument). Python methods have an explicit self-parameter to access [instance data](https://en.wikipedia.org/wiki/Instance_data), in contrast to the implicit self (or this) in some other object-oriented programming languages (e.g., [C++](https://en.wikipedia.org/wiki/C%2B%2B), [Java](https://en.wikipedia.org/wiki/Java_(programming_language)), [Objective-C](https://en.wikipedia.org/wiki/Objective-C), or [Ruby](https://en.wikipedia.org/wiki/Ruby_(programming_language))).

### TYPING

Python uses [duck typing](https://en.wikipedia.org/wiki/Duck_typing) and has typed objects but untyped variable names. Type constraints are not checked at [compile time](https://en.wikipedia.org/wiki/Compile_time); rather, operations on an object may fail, signifying that the given object is not of a suitable type. Despite being [dynamically-typed](https://en.wikipedia.org/wiki/Type_system#Dynamic_type_checking_and_runtime_type_information), Python is [strongly-typed](https://en.wikipedia.org/wiki/Strong_and_weak_typing), forbidding operations that are not well-defined (for example, adding a number to a string) rather than silently attempting to make sense of them.

Python allows programmers to define their own types using [classes](https://en.wikipedia.org/wiki/Class_(computer_science)), which are most often used for [object-oriented programming](https://en.wikipedia.org/wiki/Object-oriented_programming). New [instances](https://en.wikipedia.org/wiki/Object_(computer_science)) of classes are constructed by calling the class (for example, SpamClass() or EggsClass()), and the classes are instances of the [metaclass](https://en.wikipedia.org/wiki/Metaclass) type (itself an instance of itself), allowing [metaprogramming](https://en.wikipedia.org/wiki/Metaprogramming) and [reflection](https://en.wikipedia.org/wiki/Reflection_(computer_science)).

Before version 3.0, Python had two kinds of classes: old-style and new-style.[[99]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-classy-99) The syntax of both styles is the same, the difference being whether the class object is inherited from, directly or indirectly (all new-style classes inherit from object and are instances of type). In versions of Python 2 from Python 2.2 onwards, both kinds of classes can be used. Old-style classes were eliminated in Python 3.0.

The long-term plan is to support [gradual typing](https://en.wikipedia.org/wiki/Gradual_typing) and from Python 3.5, the syntax of the language allows specifying static types but they are not checked in the default implementation, [CPython](https://en.wikipedia.org/wiki/CPython). An experimental optional static type checker named mypy supports compile-time type checking.

## LIBRARIES

Python's large [standard library](https://en.wikipedia.org/wiki/Standard_library), commonly cited as one of its greatest strengths, provides tools suited to many tasks. For Internet-facing applications, many standard formats and protocols such as [MIME](https://en.wikipedia.org/wiki/MIME) and [HTTP](https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol) are supported. It includes modules for creating [graphical user interfaces](https://en.wikipedia.org/wiki/Graphical_user_interface), connecting to [relational databases](https://en.wikipedia.org/wiki/Relational_database), [generating pseudorandom numbers](https://en.wikipedia.org/wiki/Pseudorandom_number_generator), arithmetic with arbitrary-precision decimals, manipulating [regular expressions](https://en.wikipedia.org/wiki/Regular_expression), and [unit testing](https://en.wikipedia.org/wiki/Unit_testing).

Some parts of the standard library are covered by specifications (for example, the [Web Server Gateway Interface](https://en.wikipedia.org/wiki/Web_Server_Gateway_Interface) (WSGI) implementation wsgiref follows PEP 333), but most modules are not. They are specified by their code, internal documentation, and [test suites](https://en.wikipedia.org/wiki/Test_suite). However, because most of the standard library is cross-platform Python code, only a few modules need altering or rewriting for variant implementations.

As of March 2021, the [Python Package Index](https://en.wikipedia.org/wiki/Python_Package_Index) (PyPI), the official repository for third-party Python software, contains over 290,000 packages with a wide range of functionality, including:

* [Automation](https://en.wikipedia.org/wiki/Automation)
* [Data analytics](https://en.wikipedia.org/wiki/Data_analytics)
* [Databases](https://en.wikipedia.org/wiki/Databases)
* [Documentation](https://en.wikipedia.org/wiki/Documentation)
* [Graphical user interfaces](https://en.wikipedia.org/wiki/Graphical_user_interfaces)
* [Image processing](https://en.wikipedia.org/wiki/Image_processing)
* [Machine learning](https://en.wikipedia.org/wiki/Machine_learning)
* [Mobile App](https://en.wikipedia.org/wiki/Mobile_App)
* [Multimedia](https://en.wikipedia.org/wiki/Multimedia)
* [Computer Networking](https://en.wikipedia.org/wiki/Computer_networking)
* [Scientific computing](https://en.wikipedia.org/wiki/Scientific_computing)
* [System administration](https://en.wikipedia.org/wiki/System_administration)
* [Test frameworks](https://en.wikipedia.org/wiki/Test_framework)
* [Text processing](https://en.wikipedia.org/wiki/Text_processing)
* [Web frameworks](https://en.wikipedia.org/wiki/Web_framework)
* [Web scraping](https://en.wikipedia.org/wiki/Web_scraping)

## DEVELOPMENT ENVIRONMENTS

Most Python implementations (including CPython) include a [read–eval–print loop](https://en.wikipedia.org/wiki/Read%E2%80%93eval%E2%80%93print_loop) (REPL), permitting them to function as a [command line interpreter](https://en.wikipedia.org/wiki/Command_line_interpreter) for which the user enters statements sequentially and receives results immediately.

Other shells, including [IDLE](https://en.wikipedia.org/wiki/IDLE) and [IPython](https://en.wikipedia.org/wiki/IPython), add further abilities such as improved auto-completion, session state retention and [syntax highlighting](https://en.wikipedia.org/wiki/Syntax_highlighting).

As well as standard desktop [integrated development environments](https://en.wikipedia.org/wiki/Integrated_development_environment), there are [Web browser](https://en.wikipedia.org/wiki/Web_browser)-based IDEs; [SageMath](https://en.wikipedia.org/wiki/SageMath) (intended for developing science and math-related Python programs); [PythonAnywhere](https://en.wikipedia.org/wiki/PythonAnywhere), a browser-based IDE and hosting environment; and Canopy IDE, a commercial Python IDE emphasizing [scientific computing](https://en.wikipedia.org/wiki/Scientific_computing).[[120]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-120)

## IMPLEMENTATIONS

### REFERENCE IMPLEMENTATION

[CPython](https://en.wikipedia.org/wiki/CPython) is the [reference implementation](https://en.wikipedia.org/wiki/Reference_implementation) of Python. It is written in [C](https://en.wikipedia.org/wiki/C_(programming_language)), meeting the [C89](https://en.wikipedia.org/wiki/C89_(C_version)) standard with several select [C99](https://en.wikipedia.org/wiki/C99) features (with later C versions out, it's considered outdated; CPython includes its own C extensions, but third-party extensions are not limited to older C versions, can e.g. be implemented with [C11](https://en.wikipedia.org/wiki/C11_(C_standard_revision)) or [C++](https://en.wikipedia.org/wiki/C%2B%2B)). It compiles Python programs into an intermediate [bytecode](https://en.wikipedia.org/wiki/Bytecode) which is then executed by its [virtual machine](https://en.wikipedia.org/wiki/Virtual_machine).[[126]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-68-126) CPython is distributed with a large standard library written in a mixture of C and native Python. It is available for many platforms, including [Windows](https://en.wikipedia.org/wiki/Microsoft_Windows) (starting with Python 3.9, the Python installer deliberately fails to install on [Windows 7](https://en.wikipedia.org/wiki/Windows_7) and 8; [Windows XP](https://en.wikipedia.org/wiki/Windows_XP) was supported until Python 3.5) and most modern [Unix-like](https://en.wikipedia.org/wiki/Unix-like) systems, including macOS (and [Apple M1](https://en.wikipedia.org/wiki/Apple_M1) Macs, since Python 3.9.1, with experimental installer) and unofficial support for e.g. [VMS](https://en.wikipedia.org/wiki/OpenVMS). Platform portability was one of its earliest priorities, during the Python 1 and 2 time-frame, even [OS/2](https://en.wikipedia.org/wiki/OS/2) and [Solaris](https://en.wikipedia.org/wiki/Solaris_(operating_system)) were supported; support has since been dropped for a lot of platforms.

### OTHER IMPLEMENTATIONS

* [PyPy](https://en.wikipedia.org/wiki/PyPy) is a fast, compliant interpreter of Python 2.7 and 3.6. Its [just-in-time compiler](https://en.wikipedia.org/wiki/Just-in-time_compilation) brings a significant speed improvement over CPython but several libraries written in C cannot be used with it.
* [Stackless Python](https://en.wikipedia.org/wiki/Stackless_Python) is a significant fork of CPython that implements [microthreads](https://en.wikipedia.org/wiki/Microthread); it does not use the [call stack](https://en.wikipedia.org/wiki/Call_stack) in the same way, thus allowing massively concurrent programs. PyPy also has a stackless version.
* [MicroPython](https://en.wikipedia.org/wiki/MicroPython) and [CircuitPython](https://en.wikipedia.org/wiki/CircuitPython) are Python 3 variants optimized for [microcontrollers](https://en.wikipedia.org/wiki/Microcontroller), including [Lego Mindstorms EV3](https://en.wikipedia.org/wiki/Lego_Mindstorms_EV3).
* Pyston is a variant of the Python runtime that uses just-in-time compilation to speed up the execution of Python programs.

### UNSUPPORTED IMPLEMENTATIONS

Other just-in-time Python compilers have been developed, but are now unsupported:

* Google began a project named [Unladen Swallow](https://en.wikipedia.org/wiki/Unladen_Swallow) in 2009, with the aim of speeding up the Python interpreter fivefold by using the [LLVM](https://en.wikipedia.org/wiki/LLVM), and of improving its multithreading ability to scale to thousands of cores,[[138]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-74-138) while ordinary implementations suffer from the [global interpreter lock](https://en.wikipedia.org/wiki/Global_interpreter_lock).
* [Psyco](https://en.wikipedia.org/wiki/Psyco) is a discontinued [just-in-time](https://en.wikipedia.org/wiki/Just-in-time_compilation) [specializing](https://en.wikipedia.org/wiki/Run-time_algorithm_specialization) compiler (which didn't support Python 2.7 or later) that integrates with CPython and transforms bytecode to machine code at runtime. The emitted code is specialized for certain [data types](https://en.wikipedia.org/wiki/Data_type) and is faster than the standard Python code.
* [PyS60](https://en.wikipedia.org/wiki/PyS60) was a Python 2 interpreter for [Series 60](https://en.wikipedia.org/wiki/Series_60) mobile phones released by [Nokia](https://en.wikipedia.org/wiki/Nokia) in 2005. It implemented many of the modules from the standard library and some additional modules for integrating with the [Symbian](https://en.wikipedia.org/wiki/Symbian) operating system. The Nokia [N900](https://en.wikipedia.org/wiki/N900) also supports Python with [GTK](https://en.wikipedia.org/wiki/GTK) widget libraries, enabling programs to be written and run on the target device.

### CROSS-COMPILERS TO OTHER LANGUAGES

There are several compilers to high-level [object languages](https://en.wikipedia.org/wiki/Object_language), with either unrestricted Python, a restricted subset of Python, or a language similar to Python as the source language:

* [Cython](https://en.wikipedia.org/wiki/Cython) compiles (a superset of) Python 2.7 to [C](https://en.wikipedia.org/wiki/C_(programming_language)) (while the resulting code is also usable with Python 3 and also e.g. [C++](https://en.wikipedia.org/wiki/C%2B%2B)).
* [Nuitka](https://en.wikipedia.org/wiki/Nuitka) compiles Python into C++.[[140]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-140)
* Pythran compiles a subset of Python 3 to [C++](https://en.wikipedia.org/wiki/C%2B%2B).[[141]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-141)[[142]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-142)[[143]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-143)
* [Pyrex](https://en.wikipedia.org/wiki/Pyrex_(programming_language)) (latest release in 2010) and [Shed Skin](https://en.wikipedia.org/wiki/Shed_Skin) (latest release in 2013) compile to C and C++ respectively.
* Google's Grumpy (latest release in 2017) [transpiles](https://en.wikipedia.org/wiki/Transpile) Python 2 to [Go](https://en.wikipedia.org/wiki/Go_(programming_language)).[[144]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-144)[[145]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-145)[[146]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-146)
* [IronPython](https://en.wikipedia.org/wiki/IronPython) (now abandoned by Microsoft) allows running Python 2.7 programs on the .NET [Common Language Runtime](https://en.wikipedia.org/wiki/Common_Language_Runtime).
* [Jython](https://en.wikipedia.org/wiki/Jython) compiles Python 2.7 to Java bytecode, allowing the use of the Java libraries from a Python program.
* [MyHDL](https://en.wikipedia.org/wiki/MyHDL) is a Python-based [hardware description language](https://en.wikipedia.org/wiki/Hardware_description_language) (HDL), that converts MyHDL code to [Verilog](https://en.wikipedia.org/wiki/Verilog) or [VHDL](https://en.wikipedia.org/wiki/VHDL) code.
* [Numba](https://en.wikipedia.org/wiki/Numba) uses [LLVM](https://en.wikipedia.org/wiki/LLVM) to compile a subset of Python to machine code.
* Brython,[[147]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-147) Transcrypt[[148]](https://en.wikipedia.org/wiki/Python_(programming_language)" \l "cite_note-148)[[149]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-149) and [Pyjs](https://en.wikipedia.org/wiki/Pyjs) (latest release in 2012) compile Python to [JavaScript](https://en.wikipedia.org/wiki/JavaScript).
* [RPython](https://en.wikipedia.org/wiki/RPython) can be compiled to [C](https://en.wikipedia.org/wiki/C_(programming_language)), and is used to build the PyPy interpreter of Python.

### PERFORMANCE

A performance comparison of various Python implementations on a non-numerical (combinatorial) workload was presented at EuroSciPy '13.[[150]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-150) Python's performance compared to other programming languages is also benchmarked by [The Computer Language Benchmarks Game](https://en.wikipedia.org/wiki/The_Computer_Language_Benchmarks_Game).

## DEVELOPMENT

Python's development is conducted largely through the Python Enhancement Proposal (PEP) process, the primary mechanism for proposing major new features, collecting community input on issues and documenting Python design decisions.[[152]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-PepCite000-152) Python coding style is covered in PEP 8. Outstanding PEPs are reviewed and commented on by the Python community and the steering council.

Enhancement of the language corresponds with development of the CPython reference implementation. The mailing list python-dev is the primary forum for the language's development. Specific issues are discussed in the [Roundup](https://en.wikipedia.org/wiki/Roundup_(issue_tracker)) [bug tracker](https://en.wikipedia.org/wiki/Bug_tracker) hosted at [bugs.python.org](https://bugs.python.org). Development originally took place on a [self-hosted](https://en.wikipedia.org/wiki/Self-hosting_(web_services)) source-code repository running [Mercurial](https://en.wikipedia.org/wiki/Mercurial), until Python moved to [GitHub](https://en.wikipedia.org/wiki/GitHub) in January 2017.

CPython's public releases come in three types, distinguished by which part of the version number is incremented:

* Backward-incompatible versions, where code is expected to break and needs to be manually [ported](https://en.wikipedia.org/wiki/Ported). The first part of the version number is incremented. These releases happen infrequently—version 3.0 was released 8 years after 2.0.
* Major or "feature" releases, occurred about every 18 months but with the adoption of a yearly release cadence starting with Python 3.9 are expected to happen once a year. They are largely compatible but introduce new features. The second part of the version number is incremented. Each major version is supported by bugfixes for several years after its release.
* Bugfix releases, which introduce no new features, occur about every 3 months and are made when a sufficient number of bugs have been fixed upstream since the last release. Security vulnerabilities are also patched in these releases. The third and final part of the version number is incremented.

Many [alpha, beta, and release-candidates](https://en.wikipedia.org/wiki/Beta_release) are also released as previews and for testing before final releases. Although there is a rough schedule for each release, they are often delayed if the code is not ready. Python's development team monitors the state of the code by running the large [unit test](https://en.wikipedia.org/wiki/Unit_test) suite during development.

The major [academic conference](https://en.wikipedia.org/wiki/Academic_conference) on Python is [PyCon](https://en.wikipedia.org/wiki/PyCon). There are also special Python mentoring programmes, such as [Pyladies](https://en.wikipedia.org/wiki/Pyladies).

Pythons 3.10 deprecates wstr (to be removed in Python 3.12; meaning Python extensionsneed to be modified by then),[[162]](https://en.wikipedia.org/wiki/Python_(programming_language)" \l "cite_note-162) and also plans to add [pattern matching](https://en.wikipedia.org/wiki/Pattern_matching) to the language.

## USES

Since 2003, Python has consistently ranked in the top ten most popular programming languages in the [TIOBE Programming Community Index](https://en.wikipedia.org/wiki/TIOBE_Programming_Community_Index) where, as of February 2021, it is the third most popular language (behind [Java](https://en.wikipedia.org/wiki/Java_(programming_language)), and [C](https://en.wikipedia.org/wiki/C_(programming_language))). It was selected Programming Language of the Year (for "the highest rise in ratings in a year") in 2007, 2010, 2018, and 2020 (the only language to do so four times).

An empirical study found that scripting languages, such as Python, are more productive than conventional languages, such as C and Java, for programming problems involving string manipulation and search in a dictionary, and determined that memory consumption was often "better than Java and not much worse than C or C++".[[172]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-AutoNT-28-172)

Large organizations that use Python include [Wikipedia](https://en.wikipedia.org/wiki/Wikipedia), [Google](https://en.wikipedia.org/wiki/Google), [Yahoo!](https://en.wikipedia.org/wiki/Yahoo!), [CERN](https://en.wikipedia.org/wiki/CERN), [NASA](https://en.wikipedia.org/wiki/NASA), [Facebook](https://en.wikipedia.org/wiki/Facebook) [Amazon](https://en.wikipedia.org/wiki/Amazon_(company)), [Instagram](https://en.wikipedia.org/wiki/Instagram), [Spotify](https://en.wikipedia.org/wiki/Spotify) and some smaller entities like [ILM](https://en.wikipedia.org/wiki/Industrial_Light_%26_Magic) and [ITA](https://en.wikipedia.org/wiki/ITA_Software). The social news networking site [Reddit](https://en.wikipedia.org/wiki/Reddit) was written mostly in Python.

Python can serve as a [scripting language](https://en.wikipedia.org/wiki/Scripting_language) for [web applications](https://en.wikipedia.org/wiki/Web_application), e.g., via [mod\_wsgi](https://en.wikipedia.org/wiki/Mod_wsgi) for the [Apache web server](https://en.wikipedia.org/wiki/Apache_web_server). With [Web Server Gateway Interface](https://en.wikipedia.org/wiki/Web_Server_Gateway_Interface), a standard API has evolved to facilitate these applications. [Web frameworks](https://en.wikipedia.org/wiki/Web_framework) like [Django](https://en.wikipedia.org/wiki/Django_(web_framework)), [Pylons](https://en.wikipedia.org/wiki/Pylons_(web_framework)), [Pyramid](https://en.wikipedia.org/wiki/Pyramid_(web_framework)), [TurboGears](https://en.wikipedia.org/wiki/TurboGears), [web2py](https://en.wikipedia.org/wiki/Web2py), [Tornado](https://en.wikipedia.org/wiki/Tornado_(web_server)), [Flask](https://en.wikipedia.org/wiki/Flask_(web_framework)), [Bottle](https://en.wikipedia.org/wiki/Bottle_(web_framework)) and [Zope](https://en.wikipedia.org/wiki/Zope) support developers in the design and maintenance of complex applications. [Pyjs](https://en.wikipedia.org/wiki/Pyjs) and [IronPython](https://en.wikipedia.org/wiki/IronPython) can be used to develop the client-side of Ajax-based applications. [SQLAlchemy](https://en.wikipedia.org/wiki/SQLAlchemy) can be used as a [data mapper](https://en.wikipedia.org/wiki/Data_mapper_pattern) to a relational database. [Twisted](https://en.wikipedia.org/wiki/Twisted_(software)) is a framework to program communications between computers, and is used (for example) by [Dropbox](https://en.wikipedia.org/wiki/Dropbox_(service)).

Libraries such as [NumPy](https://en.wikipedia.org/wiki/NumPy), [SciPy](https://en.wikipedia.org/wiki/SciPy) and [Matplotlib](https://en.wikipedia.org/wiki/Matplotlib) allow the effective use of Python in [scientific computing](https://en.wikipedia.org/wiki/Scientific_computing),[[184]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-cise-184)[[185]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-millman-185) with specialized libraries such as [Biopython](https://en.wikipedia.org/wiki/Biopython) and [Astropy](https://en.wikipedia.org/wiki/Astropy) providing domain-specific functionality. [SageMath](https://en.wikipedia.org/wiki/SageMath) is a [mathematical software](https://en.wikipedia.org/wiki/Mathematical_software) with a [notebook interface](https://en.wikipedia.org/wiki/Notebook_interface) programmable in Python: its library covers many aspects of [mathematics](https://en.wikipedia.org/wiki/Mathematics), including [algebra](https://en.wikipedia.org/wiki/Algebra), [combinatorics](https://en.wikipedia.org/wiki/Combinatorics), [numerical mathematics](https://en.wikipedia.org/wiki/Numerical_mathematics), [number theory](https://en.wikipedia.org/wiki/Number_theory), and [calculus](https://en.wikipedia.org/wiki/Calculus). [OpenCV](https://en.wikipedia.org/wiki/OpenCV) has python bindings with a rich set of features for [computer vision](https://en.wikipedia.org/wiki/Computer_vision) and [image processing](https://en.wikipedia.org/wiki/Digital_image_processing).

Python is commonly used in [artificial intelligence](https://en.wikipedia.org/wiki/Artificial_intelligence) projects and machine learning projects with the help of libraries like [TensorFlow](https://en.wikipedia.org/wiki/TensorFlow), [Keras](https://en.wikipedia.org/wiki/Keras), [Pytorch](https://en.wikipedia.org/wiki/PyTorch) and [Scikit-learn](https://en.wikipedia.org/wiki/Scikit-learn). As a scripting language with [modular architecture](https://en.wikipedia.org/wiki/Modular_programming), simple syntax and rich text processing tools, Python is often used for [natural language processing](https://en.wikipedia.org/wiki/Natural_language_processing).

Python has been successfully embedded in many software products as a scripting language, including in [finite element method](https://en.wikipedia.org/wiki/Finite_element_method) software such as [Abaqus](https://en.wikipedia.org/wiki/Abaqus), 3D parametric modeler like [FreeCAD](https://en.wikipedia.org/wiki/FreeCAD), 3D animation packages such as [3ds Max](https://en.wikipedia.org/wiki/3ds_Max), [Blender](https://en.wikipedia.org/wiki/Blender_(software)), [Cinema 4D](https://en.wikipedia.org/wiki/Cinema_4D), [Lightwave](https://en.wikipedia.org/wiki/Lightwave), [Houdini](https://en.wikipedia.org/wiki/Houdini_(software)), [Maya](https://en.wikipedia.org/wiki/Maya_(software)), [modo](https://en.wikipedia.org/wiki/Modo_(software)), [MotionBuilder](https://en.wikipedia.org/wiki/MotionBuilder), [Softimage](https://en.wikipedia.org/wiki/Autodesk_Softimage), the visual effects compositor [Nuke](https://en.wikipedia.org/wiki/Nuke_(software)), 2D imaging programs like [GIMP](https://en.wikipedia.org/wiki/GIMP), [Inkscape](https://en.wikipedia.org/wiki/Inkscape), [Scribus](https://en.wikipedia.org/wiki/Scribus) and [Paint Shop Pro](https://en.wikipedia.org/wiki/Paint_Shop_Pro), and [musical notation](https://en.wikipedia.org/wiki/Musical_notation) programs like [scorewriter](https://en.wikipedia.org/wiki/Scorewriter) and [capella](https://en.wikipedia.org/wiki/Capella_(notation_program)). [GNU Debugger](https://en.wikipedia.org/wiki/GNU_Debugger) uses Python as a [pretty printer](https://en.wikipedia.org/wiki/Prettyprint) to show complex structures such as C++ containers. [Esri](https://en.wikipedia.org/wiki/Esri) promotes Python as the best choice for writing scripts in [ArcGIS](https://en.wikipedia.org/wiki/ArcGIS). It has also been used in several video games, and has been adopted as first of the three available [programming languages](https://en.wikipedia.org/wiki/Programming_language) in [Google App Engine](https://en.wikipedia.org/wiki/Google_App_Engine), the other two being [Java](https://en.wikipedia.org/wiki/Java_(software_platform)) and [Go](https://en.wikipedia.org/wiki/Go_(programming_language)).

Many operating systems include Python as a standard component. It ships with most [Linux distributions](https://en.wikipedia.org/wiki/Linux_distribution),[[199]](https://en.wikipedia.org/wiki/Python_(programming_language)#cite_note-199) [AmigaOS 4](https://en.wikipedia.org/wiki/AmigaOS_4) (using Python 2.7), [FreeBSD](https://en.wikipedia.org/wiki/FreeBSD) (as a package), [NetBSD](https://en.wikipedia.org/wiki/NetBSD), [OpenBSD](https://en.wikipedia.org/wiki/OpenBSD) (as a package) and [macOS](https://en.wikipedia.org/wiki/MacOS) and can be used from the command line (terminal). Many Linux distributions use installers written in Python: [Ubuntu](https://en.wikipedia.org/wiki/Ubuntu_(operating_system)) uses the [Ubiquity](https://en.wikipedia.org/wiki/Ubiquity_(software)) installer, while [Red Hat Linux](https://en.wikipedia.org/wiki/Red_Hat_Linux) and [Fedora](https://en.wikipedia.org/wiki/Fedora_(operating_system)) use the [Anaconda](https://en.wikipedia.org/wiki/Anaconda_(installer)) installer. [Gentoo Linux](https://en.wikipedia.org/wiki/Gentoo_Linux) uses Python in its [package management system](https://en.wikipedia.org/wiki/Package_management_system), [Portage](https://en.wikipedia.org/wiki/Portage_(software)).

Python is used extensively in the [information security](https://en.wikipedia.org/wiki/Information_security) industry, including in exploit development

Most of the [Sugar](https://en.wikipedia.org/wiki/Sugar_(software)) software for the [One Laptop per Child](https://en.wikipedia.org/wiki/One_Laptop_per_Child) XO, now developed at [Sugar Labs](https://en.wikipedia.org/wiki/Sugar_Labs), is written in Python. The [Raspberry Pi](https://en.wikipedia.org/wiki/Raspberry_Pi) [single-board computer](https://en.wikipedia.org/wiki/Single-board_computer) project has adopted Python as its main user-programming language.

[LibreOffice](https://en.wikipedia.org/wiki/LibreOffice) includes Python, and intends to replace Java with Python. Its Python Scripting Provider is a core feature since Version 4.0 from 7 February 2013.

**REQUIREMENT SPECIFICATION**:

### INTRODUCTION:

**Purpose:** The main purpose for preparing this document is to give a general insight into the analysis and requirements of the existing system or situation and for determining the operating characteristics of the system.

**Scope:** This Document plays a vital role in the development life cycle (SDLC) as it describes the complete requirement of the system. It is meant for use by the developers and will be the basic during testing phase. Any changes made to the requirements in the future will have to go through formal change approval process.

## Developers Responsibilities Overview:

**The developer is responsible for:**

1. Developing the system, which meets the SRS and solving all the requirements of the system?
2. Demonstrating the system and installing the system at client's location after the acceptance testing is successful.
3. Submitting the required user manual describing the system interfaces to work on it and also the documents of the system.
4. Conducting any user training that might be needed for using the system.
5. Maintaining the system for a period of one year after installation.

**Functional Requirements**:

**Input:** The major inputs for Integration of Web based Accommodation Upholding Maintenance System can be categorized module -wise. Basically all the information is managed by the software and in order to access the information one has to produce one's identity by entering the user-id and password. Every user has their own domain of access beyond which the access is dynamically refrained rather denied.

**Output:** The major outputs of the system are tables and reports. Tables are created dynamically to meet the requirements on demand. Reports, as it is obvious, carry the gist of the whole information that flows across the institution.

This application must be able to produce output at different modules for different inputs.

**GENERIC RISKS**

A risk is an unwanted event that has negative consequences. Project managers will engage in risk management to understand and control the risks on their projects. We can distinguish risks from other project events by looking for three things:

* A loss associated with the event.
* The likelihood that the event will occur.

The degree to which we can change the outcome.

The generic risks such as the Product size risk, business impact risks, Customer–Related risks, Process risks, Technology risks, Development environment risks, Security risks etc. for this project are analyzed and documented by the senior staffs in the organization. This project is developed by considering these issues and with the constant support from senior staffs in the organization.

**SECURITY TECHNOLOGIES & POLICIES**

The software quality assurance is comprised of a variety of tasks associated with seven major activities.

1. Application of technical methods.
2. Conduct of formal technical reviews
3. Software testing
4. Enforcement of standards
5. Control of change
6. Measurement
7. Record keeping and reporting

The quality begins with a set of technical methods and tools that help the analyst to achieve high quality specification and the designer to develop high-quality design.

The next activity involves assessment for quality for the design that is created which is the formal technical review.

Software testing combines a multi step strategy with a series of test case design methods that help ensure effective error detection.

It is assumed that testing will uncover most of the errors. In most cases the standards are dictated by customers, in other situations, by self-imposed.

Every change to software has the potential for introducing errors or creating side effects that propagate errors. The change control process contributes directly t software quality by formalizing request for change, evaluating the nature of change and controlling the impact of change.

An important objective of quality assurance is to track the software quality and assess the impact of methodological and procedural changes on improved software quality.

Record keeping and recording for software quality assurance provide procedures for the evaluation and separation of Software quality assurance information

**SYSTEM SECURITY**

Software integrity has become increasingly important in the age of hackers and firewalls. This attributes measures a system ability to withstand attacks (both accidental and intentional) to its security. Attacks can be made on all three components of software program, data, and documents

To measure integrity, two additional attributes must be defined

* Thread
* Security

**THREAD**

Threat is the probability (which can be derived or estimated from empirical evidence) that an attack of specific type occur within a specific time.

**Security**

Security is the probability (which can be estimated or derived from empirical evidence) that attack on the specific type will be repelled.

**Security Testing**

Any computer based system that manages sensitive information or causes action that can improperly harm(or benefit) individuals is the target for improper or illegal penetration. Penetration spans a board range of activities; hackers who penetrate system for sport; disgruntled employee who attempt to penetrate for revenge; dishonest individual who penetrate for illicit personnel gains

Security testing to verify that protection mechanism built into a system will in fact provide proper protection form improper penetration. During system testing, the tester plays the role of the individual who desires to penetrate the system. Anything goes! The tester may attempt to acquire password through external clerical means; may attack the system with custom software designed to break down any defenses that have been constructed may overwhelm the system thereby denying the service to other; may purposely cause system errors.

**Feasibility Report:**

**TECHINICAL FEASIBILITY**:

Evaluating the technical feasibility is the trickiest part of a feasibility study. This is because, at this point in time, not too many detailed design of the system, making it difficult to access issues like performance, costs on (on account of the kind of technology to be deployed) etc. A number of issues have to be considered while doing a technical analysis.

Understand the different technologies involved in the proposed system:

Before commencing the project, we have to be very clear about what are the technologies that are to be required for the development of the new system.

Find out whether the organization currently possesses the required technologies:

Is the required technology available with the organization?

If so is the capacity sufficient?

For instance –

“Will the current printer be able to handle the new reports and forms required for the new system?”

**OPERATIONAL FEASIBILITY**:

Proposed projects are beneficial only if they can be turned into information systems that will meet the organizations operating requirements. Simply stated, this test of feasibility asks if the system will work when it is developed and installed. Are there major barriers to Implementation? Here are questions that will help test the operational feasibility of a project:

Is there sufficient support for the project from management from users? If the current system is well liked and used to the extent that persons will not be able to see reasons for change, there may be resistance.

Are the current business methods acceptable to the user? If they are not, Users may welcome a change that will bring about a more operational and useful systems.

Have the user been involved in the planning and development of the project?

Early involvement reduces the chances of resistance to the system and in

General and increases the likelihood of successful project.

Since the proposed system was to help reduce the hardships encountered. In the existing manual system, the new system was considered to be operational feasible.

**ECONOMIC FEASIBILITY**:

Economic feasibility attempts 2 weigh the costs of developing and implementing a new system, against the benefits that would accrue from having the new system in place. This feasibility study gives the top management the economic justification for the new system.

A simple economic analysis which gives the actual comparison of costs and benefits are much more meaningful in this case. In addition, this proves to be a useful point of reference to compare actual costs as the project progresses. There could be various types of intangible benefits on account of automation. These could include increased customer satisfaction, improvement in product quality better decision making timeliness of information, expediting activities, improved accuracy of operations, better documentation and record keeping, faster retrieval of information, better employee morale.

**TESTING AND DEBUGING TECHNIQUES**

**Testing Techniques:**

**SOFTWARE TESTING**

**GENERAL**

Testing is the most important phases in the software development activity. In software development life cycle (SDLC), the main aim of the testing process in the quality, the developed software is tested against attaining the required functionality and performance. During the testing process the software is worked with some particular test case and the output of the test cases are analyzed whether the software is working according to the expectations or not.

The success of the testing process in determining the error is mostly depends upon the test case criteria, for testing any software we need to have a description of the expected behavior of the system and method of determining whether the observed behavior confirmed to the expected behavior.

**DEVELOPING METHODOLOGIES**

The test process is initiated by developing a comprehensive plan to test the general functionality and special features on a variety of platform combinations. Strict quality control procedures are used.

The process verifies that the application meets the requirements specified in the system requirements document and is bug free. The following are the considerations used to develop the framework from developing the testing methodologies.

**TYPES OF TEST**

Since the error in the software can be injured at any stage. so, we have carry out the testing process at different levels during the development. The basic levels of testing are

* + - Unit Testing
    - Integration Testing
    - System Testing
    - Acceptance Testing.

**Unit testing**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program input produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

**Functional test**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

**System Test**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

**Performance Test**

The Performance test ensures that the output be produced within the time limits,and the time taken by the system for compiling, giving response to the users and request being send to the system for to retrieve the results.

**Integration Testing**

Integration testing is a systematic technique for constructing the program structure, while at the same time conducting tests to uncover error associated with interfacing. The following are the types of Integration Testing

* Top down Integration
* Bottom-up Integration

**Top down Integration**

This method is an incremental approach to the construction of program structure.Modules are integrated by moving downward through the control hierarchy,beginning with the program module.The module subordinates to the main program module are incorporated into the structure in either a depth first or breathe first manner

**Bottom-up integration**

This method begins the construction and testing with the modules at the lowest level in the program structure.Since the modules are integrated from the bottom up,processing required for modules subordinate to a given level is always available and the need for stubs is eliminated.The bottom up integration strategy may be implemented with the following steps:

* Low modules were combined into clusters that perform specific software sub function
* The clusters were tested Drives are removed and clusters are combined moving upward in the program structure

**Acceptance Testing**

Once the application is ready to be released the crucial step is User Acceptance Testing. In this step a group representing a cross section of end users tests the application.   
The user acceptance testing is done using real world scenarios and perceptions relevant to the end users. User Acceptance Testing is often the final step before rolling out the application.Usually the end users who will be using the applications test the application before ‘accepting’ the application.This type of testing gives the end users the confidence that the application being delivered to them meets their requirements.This testing also helps nail bugs related to usability of the application

**Build the test plan**

Any project can be divided into units that can be further performed for detailed processing. Then a testing strategy for each of this unit is carried out. Unit testing helps to identity the possible bugs in the individual component, so the component that has bugs can be identified and can be rectified from errors.

**QUALITY ASSURANCE**

Quality assurance defines the objectives of a project and reviews the overall activities so that error are corrected early in the development process.

**Levels of Quality Assurance**

Quality Assurance comes in three main levels namely

* Testing
* Validation
* Certification

**Testing**

In system testing a common view is to eliminate program errors. This is extremely difficult and time consuming. Since designers cannot prove 100% accuracy. A successful test, then, is one that finds errors.

**Validation**

It checks the quality of the software in both simulated and live environments. In the Simulated approach the developers test the product(Alpha test) on their workplace to make the products meet its requirements. In the Live Environment phase the product is given to the customer to evaluate the product’s functionality(Beta test).

Validation refers to the different set of activities that ensure that software correctly implements a specific function and the software that been built is traceable to customer requirements.

Software validation is achieved through a series of black-box test that demonstrate conformity with requirement. After each validation check a test has been conducted, one of the two possible condition exists

* The function or performance characteristics conform to specification and are expected
* A deviation from specification is uncovered and a deficiency list is created

**Alpha and Beta testing**

The alpha testing is conducted at the developer’s site by the customer. the software is used in the natural setting with the developer “looking over the developer” and recording errors and usages problems. Alpha test is conducted in controlled environment

The beta testing is conducted at one or more customer site by the end user of the software. Unlike software testing the developer is generally not present. Therefore beta test is live application of the software in an environment that cannot be controlled by the developer

**Validation check applied in the project**

* The files entered in the project must only have the doc extension
* The data entered must have only one format
* One should not make a enter into without checking the password
* The qc must enter into the status column only accepted or not accepted
* File which are already be registered should not be registered once again or other time
* Qc can’t enter the file which are not entered by the dc
* Qc should enter the file into corresponding filenames registered by the dc
* Files which are registered and not accepted for the first time should not be registered again but the reentry of date and status should be done only

**Certification**

Certification is to certify that the program or software package is correct and confirms to standards. With growing trend towards purchasing ready to use software, certification has become more important.

**GENERIC RISKS**

A risk is an unwanted event that has negative consequences. Project managers will engage in risk management to understand and control the risks on their projects. We can distinguish risks from other project events by looking for three things:

* A loss associated with the event.
* The likelihood that the event will occur.

The degree to which we can change the outcome.

The generic risks such as the Product size risk, business impact risks, Customer–Related risks, Process risks, Technology risks, Development environment risks, Security risks etc. for this project are analyzed and documented by the senior staffs in the organization. This project is developed by considering these issues and with the constant support from senior staffs in the organization.

**SECURITY TECHNOLOGIES & POLICIES**

The software quality assurance is comprised of a variety of tasks associated with seven major activities.

1. Application of technical methods.
2. Conduct of formal technical reviews
3. Software testing
4. Enforcement of standards
5. Control of change
6. Measurement
7. Record keeping and reporting

The quality begins with a set of technical methods and tools that help the analyst to achieve high quality specification and the designer to develop high-quality design.

The next activity involves assessment for quality for the design that is created which is the formal technical review.

Software testing combines a multi step strategy with a series of test case design methods that help ensure effective error detection.

It is assumed that testing will uncover most of the errors. In most cases the standards are dictated by customers, in other situations, by self-imposed.

Every change to software has the potential for introducing errors or creating side effects that propagate errors. The change control process contributes directly t software quality by formalizing request for change, evaluating the nature of change and controlling the impact of change.

An important objective of quality assurance is to track the software quality and assess the impact of methodological and procedural changes on improved software quality.

Record keeping and recording for software quality assurance provide procedures for the evaluation and separation of Software quality assurance information

**SYSTEM SECURITY**

Software integrity has become increasingly important in the age of hackers and firewalls. This attributes measures a system ability to withstand attacks (both accidental and intentional) to its security. Attacks can be made on all three components of software program, data, and documents

To measure integrity, two additional attributes must be defined

* Thread
* Security

**THREAD**

Threat is the probability (which can be derived or estimated from empirical evidence) that an attack of specific type occur within a specific time.

**Security**

Security is the probability (which can be estimated or derived from empirical evidence) that attack on the specific type will be repelled.

**Security Testing**

Any computer based system that manages sensitive information or causes action that can improperly harm(or benefit) individuals is the target for improper or illegal penetration. Penetration spans a board range of activities; hackers who penetrate system for sport; disgruntled employee who attempt to penetrate for revenge; dishonest individual who penetrate for illicit personnel gains

Security testing to verify that protection mechanism built into a system will in fact provide proper protection form improper penetration. During system testing, the tester plays the role of the individual who desires to penetrate the system. Anything goes! The tester may attempt to acquire password through external clerical means; may attack the system with custom software designed to break down any defenses that have been constructed may overwhelm the system thereby denying the service to other; may purposely cause system errors.

**Feasibility Report:**

**TECHINICAL FEASIBILITY**:

Evaluating the technical feasibility is the trickiest part of a feasibility study. This is because, at this point in time, not too many detailed design of the system, making it difficult to access issues like performance, costs on (on account of the kind of technology to be deployed) etc. A number of issues have to be considered while doing a technical analysis.

Understand the different technologies involved in the proposed system:

Before commencing the project, we have to be very clear about what are the technologies that are to be required for the development of the new system.

Find out whether the organization currently possesses the required technologies:

Is the required technology available with the organization?

If so is the capacity sufficient?

For instance –

“Will the current printer be able to handle the new reports and forms required for the new system?”

**OPERATIONAL FEASIBILITY**:

Proposed projects are beneficial only if they can be turned into information systems that will meet the organizations operating requirements. Simply stated, this test of feasibility asks if the system will work when it is developed and installed. Are there major barriers to Implementation? Here are questions that will help test the operational feasibility of a project:

Is there sufficient support for the project from management from users? If the current system is well liked and used to the extent that persons will not be able to see reasons for change, there may be resistance.

Are the current business methods acceptable to the user? If they are not, Users may welcome a change that will bring about a more operational and useful systems.

Have the user been involved in the planning and development of the project?

Early involvement reduces the chances of resistance to the system and in

General and increases the likelihood of successful project.

Since the proposed system was to help reduce the hardships encountered. In the existing manual system, the new system was considered to be operational feasible.

**ECONOMIC FEASIBILITY**:

Economic feasibility attempts 2 weigh the costs of developing and implementing a new system, against the benefits that would accrue from having the new system in place. This feasibility study gives the top management the economic justification for the new system.

A simple economic analysis which gives the actual comparison of costs and benefits are much more meaningful in this case. In addition, this proves to be a useful point of reference to compare actual costs as the project progresses. There could be various types of intangible benefits on account of automation. These could include increased customer satisfaction, improvement in product quality better decision making timeliness of information, expediting activities, improved accuracy of operations, better documentation and record keeping, faster retrieval of information, better employee morale.

**CONCLUSION:**

In this investigation, a proficient AI based conclusion framework has been produced for the analysis of heart infection. AI classifiers incorporate LR, K-NN, ANN, SVM, NB, and DT are utilized in the planning of the framework. Four standard element determination calculations including Relief, MRMR, LASSO, LLBFS, and proposed a novel highlight determination calculation FCMIM used to comprehend include determination issue. LOSO cross-approval strategy is utilized in the framework for the best hyper parameters determination. The framework is tried on Cleveland heart illness dataset.

**FUTURE WORK:**

Later on, we will utilize different highlights determination calculations, improvement strategies to additional expansion the exhibition of a prescient framework for HD conclusion. The controlling and treatment of infection is importance after determination, subsequently, I will chip away at treatment and recuperation of infections in future likewise for basic sickness, for example, heart, bosom, Parkinson, diabetes.

**REFERENCES**

[1] V. Y. Dombrovskiy, A. A. Martin, J. Sunderram, and H. L. Paz, “Rapid increase in hospitalization and mortality rates for severe sepsis in the united states: a trend analysis from 1993 to 2003,” Critical care medicine, vol. 35, no. 5, pp. 1244–1250, 2007.

[2] P. E. Marik and M. Mohedin, “The contrasting effects of dopamine and norepinephrine on systemic and splanchnic oxygen utilization in hyperdynamic sepsis,” Jama, vol. 272, no. 17, pp. 1354–1357, 1994.

[3] A. Rhodes, L. E. Evans, W. Alhazzani, M. M. Levy, M. Antonelli, R. Ferrer, A. Kumar, J. E. Sevransky, C. L. Sprung, M. E. Nunnally et al., “Surviving sepsis campaign: international guidelines for management of sepsis and septic shock: 2016,” Intensive Care Medicine, vol. 43, no. 3, pp. 304–377, 2017.

[4] H. Beloeil, J.-X. Mazoit, D. Benhamou, and J. Duranteau, “Nore- pinephrine kinetics and dynamics in septic shock and trauma patients,” British Journal of Anaesthesia, vol. 95, no. 6, pp. 782–788, 2005.

[5] Y. Tang, S. Brown, J. Sorensen, and J. B. Harley, “Reduced rank least squares for real-time short term estimation of mean arterial blood pressure in septic patients receiving norepinephrine,” IEEE Journal of Translational Engineering in Health and Medicine, vol. 7, pp. 1–9, 2019.

[6] A. Holzinger, C. Biemann, C. S. Pattichis, and D. B. Kell, “What do we need to build explainable ai systems for the medical domain?” arXiv preprint arXiv:1712.09923, 2017.

[7] M. Merouani, B. Guignard, F. Vincent, S. W. Borron, P. Karoubi, J.- P. Fosse, Y. Cohen, C. Clec’h, E. Vicaut, C. Marbeuf-Gueye et al., “Norepinephrine weaning in septic shock patients by closed loop control based on fuzzy logic,” Critical Care, vol. 12, no. 6, p. R155, 2008.

[8] A. Koivo, D. Larnard, and R. Gray, “Digital control of mean arterial blood pressure in dogs by injecting a vasodilator drug,” Annals of biomedical engineering, vol. 9, no. 3, pp. 185–197, 1981.

[9] A. J. Koivo, “Automatic continuous-time blood pressure control in dogs by means of hypotensive drug injection,” IEEE Transactions on Biomedical Engineering, no. 10, pp. 574–581, 1980.

[10] J. M. Arnsparger, B. C. Mcinnis, J. R. Glover, and N. A. Normann, “Adaptive control of blood pressure,” IEEE Transactions on Biomedical Engineering, no. 3, pp. 168–176, 1983.