Queue Mono

This typeface was initially created for a technical reference I wrote for students. The book was typeset in Queue and I needed to show code samples so… I quickly made one weight of "Queue Mono." I thought that was the end of it, but at some point I decided to use it in my code editor. Then I started using it for email. Then, as I wrote and read for the next five years, I'd notice something that could be improved so I'd open the font, make the change, install the new version and go back to my real work. This is the result.

Thin
Light
Book
Medium
Bold
Black

DESIGNER
TAL LEMING
2014-?

6 STYLES
6 WEIGHTS
ROMAN

THE MONOSPACED VERSION OF QUEUE was created for use in the OpenType cookbook.

THE MONOSPACED VERSION OF QUEUE was created for use in the OpenType cookbook.

THE MONOSPACED VERSION OF QUEUE was created for use in the OpenType cookbook.

THE MONOSPACED VERSION OF QUEUE was created for use in the OpenType cookbook.

THE MONOSPACED VERSION OF QUEUE was created for use in the OpenType cookbook.

THE MONOSPACED VERSION OF QUEUE was created for use in the OpenType cookbook.

IN COMPUTER SCIENCE, A QUEUE IS A COLLECTION OF ENTITIES THAT ARE MAINTAINED IN A SEQUENCE AND CAN BE MODIFIED BY THE ADDITION OF ENTITIES AT ONE END OF THE SEQUENCE AND THE REMOVAL OF ENTITIES FROM THE OTHER END OF THE SEQUENCE. BY CONVENTION, THE END OF THE SEQUENCE AT WHICH ELEMENTS ARE ADDED IS CALLED

THE OPERATION OF ADDING AN ELEMENT TO THE REAR OF THE QUEUE IS KNOWN AS ENQUEUE, AND THE OPERATION OF REMOVING AN ELEMENT FROM THE FRONT IS KNOWN AS DEQUEUE. OTHER OPERATIONS MAY ALSO BE ALLOWED, OFTEN INCLUDING A PEEK OR FRONT OPERATION THAT RETURNS THE VALUE OF THE NEXT ELEMENT TO BE DEQUEUED WITHOUT DEQUEUING IT. THE OPERATIONS OF A QUEUE MAKE IT A FIRST IN FIRST OUT (FIFO) DATA

In computer science, a queue is a collection of entities that are maintained in a sequence and can be modified by the addition of entities at one end of the sequence and the removal of entities from the other end of the sequence. By convention, the end of the sequence at which elements are added is called

The operation of adding an element to the rear of the queue is known as enqueue, and the operation of removing an element from the front is known as dequeue. Other operations may also be allowed, often including a peek or front operation that returns the value of the next element to be dequeued without dequeuing it. The operations of a queue make it a first in first out (FIFO) data

IN COMPUTER SCIENCE, A QUEUE IS A COLLECTION OF ENTITIES THAT ARE MAINTAINED IN A SEQUENCE AND CAN BE MODIFIED BY THE ADDITION OF ENTITIES AT ONE END OF THE SEQUENCE AND THE REMOVAL OF ENTITIES FROM THE OTHER END OF THE SEQUENCE. BY CONVENTION, THE END OF THE SEQUENCE AT WHICH ELEMENTS ARE ADDED IS CALLED

THE OPERATION OF ADDING AN ELEMENT TO THE REAR OF THE QUEUE IS KNOWN AS ENQUEUE, AND THE OPERATION OF REMOVING AN ELEMENT FROM THE FRONT IS KNOWN AS DEQUEUE. OTHER OPERATIONS MAY ALSO BE ALLOWED, OFTEN INCLUDING A PEEK OR FRONT OPERATION THAT RETURNS THE VALUE OF THE NEXT ELEMENT TO BE DEQUEUED WITHOUT DEQUEUING IT. THE OPERATIONS OF A QUEUE MAKE IT A FIRST IN FIRST OUT (FIFO) DATA

In computer science, a queue is a collection of entities that are maintained in a sequence and can be modified by the addition of entities at one end of the sequence and the removal of entities from the other end of the sequence. By convention, the end of the sequence at which elements are added is called

The operation of adding an element to the rear of the queue is known as enqueue, and the operation of removing an element from the front is known as dequeue. Other operations may also be allowed, often including a peek or front operation that returns the value of the next element to be dequeued without dequeuing it. The operations of a queue make it a first in first out (FIFO) data

IN COMPUTER SCIENCE, A QUEUE IS A COLLECTION OF ENTITIES that are maintained in a sequence and can be modified by the addition of entities at one end of the sequence and the removal of entities from the other end of the sequence. By convention, the end of the sequence at which elements are added is called the back, tail, or

IN A PRIORITY QUEUE, AN ELEMENT WITH HIGH PRIORITY IS SERVED BEFORE AN ELEMENT WITH LOW PRIORITY. IN SOME IMPLEMENTATIONS, IF TWO ELEMENTS HAVE THE SAME PRIORITY, THEY ARE SERVED ACCORDING TO THE ORDER IN WHICH THEY WERE ENQUEUED, WHILE IN OTHER IMPLEMENTATIONS, ORDERING OF ELEMENTS WITH THE SAME PRIORITY IS UNDEFINED. WHILE PRIORITY QUEUES ARE OFTEN IMPLEMENTED WITH HEAPS, THEY ARE CONCEPTUALLY DISTINCT FROM HEAPS. A PRIORITY QUEUE IS A CONCEPT LIKE "A LIST" OR "A MAP"; JUST AS A LIST CAN BE IMPLEMENTED

The operation of adding an element to the rear of the queue is known as enqueue, and the operation of removing an element from the front is known as dequeue. Other operations may also be allowed, often including a peek or front operation that returns the value of the next element to be dequeued without dequeuing

In a FIFO data structure, the first element added to the queue will be the first one to be removed. This is equivalent to the requirement that once a new element is added, all elements that were added before have to be removed before the new element can be removed. A queue is an example of a linear data structure, or more abstractly a sequential collection. Queues are common in computer programs, where they are implemented as data structures coupled with access routines, as an abstract data structure or in object-oriented languages as classes. Common implementations are circular buffers and linked lists. Queues provide services in computer science, transport, and operations research where various entities such as data, objects, persons, or events are stored and held to be processed later. In these contexts, the queue performs the function of a buffer. Another usage of queues is in the implementation of breadth-first search. Theoretically, one characteristic of a queue is that it does not have a specific capacity. Regardless of how many elements

The simple trick of turning the array into a closed circle and letting the head and tail drift around endlessly in that circle makes it unnecessary to ever move items stored in the array. If n is the size of the array, then computing indices modulo n will turn the array into a circle. This is still the conceptually simplest way to construct a queue in a high-level language, but it does admittedly slow things down a little, because the array indices must be compared to zero and the array size, which is comparable to the time taken to check whether an array index is out of bounds, which some languages do, but this will certainly be the method of choice for a quick and dirty implementation, or for any high-level language that does not have pointer syntax. The array size must be declared ahead of time, but some implementations simply double the declared array size when overflow occurs. Most modern languages with objects or pointers can implement or come with libraries for dynamic lists. Such data structures may have not specified a fixed capacity limit besides memory constraints. Queue overflow results from trying to add an element onto a full queue

IN COMPUTER SCIENCE, A QUEUE IS A COLLECTION OF ENTITIES THAT ARE MAINTAINED IN A SEQUENCE AND CAN BE MODIFIED BY THE ADDITION OF ENTITIES AT ONE END OF THE SEQUENCE AND THE REMOVAL OF ENTITIES FROM THE OTHER END OF THE SEQUENCE. BY CONVENTION, THE END OF THE SEQUENCE AT WHICH ELEMENTS ARE ADDED IS CALLED

THE OPERATION OF ADDING AN ELEMENT TO THE REAR OF THE QUEUE IS KNOWN AS ENQUEUE, AND THE OPERATION OF REMOVING AN ELEMENT FROM THE FRONT IS KNOWN AS DEQUEUE. OTHER OPERATIONS MAY ALSO BE ALLOWED, OFTEN INCLUDING A PEEK OR FRONT OPERATION THAT RETURNS THE VALUE OF THE NEXT ELEMENT TO BE DEQUEUED WITHOUT DEQUEUING IT. THE OPERATIONS OF A QUEUE MAKE IT A FIRST IN FIRST OUT (FIFO) DATA

In computer science, a queue is a collection of entities that are maintained in a sequence and can be modified by the addition of entities at one end of the sequence and the removal of entities from the other end of the sequence. By convention, the end of the sequence at which elements are added is called

The operation of adding an element to the rear of the queue is known as enqueue, and the operation of removing an element from the front is known as dequeue. Other operations may also be allowed, often including a peek or front operation that returns the value of the next element to be dequeued without dequeuing it. The operations of a queue make it a first in first out (FIFO) data

IN COMPUTER SCIENCE, A QUEUE IS A COLLECTION OF ENTITIES that are maintained in a sequence and can be modified by the addition of entities at one end of the sequence and the removal of entities from the other end of the sequence. By convention, the end of the sequence at which elements are added is called the back, tail, or

IN A PRIORITY QUEUE, AN ELEMENT WITH HIGH PRIORITY IS SERVED BEFORE AN ELEMENT WITH LOW PRIORITY. IN SOME IMPLEMENTATIONS, IF TWO ELEMENTS HAVE THE SAME PRIORITY, THEY ARE SERVED ACCORDING TO THE ORDER IN WHICH THEY WERE ENQUEUED, WHILE IN OTHER IMPLEMENTATIONS, ORDERING OF ELEMENTS WITH THE SAME PRIORITY IS UNDEFINED. WHILE PRIORITY QUEUES ARE OFTEN IMPLEMENTED WITH HEAPS, THEY ARE CONCEPTUALLY DISTINCT FROM HEAPS. A PRIORITY QUEUE IS A CONCEPT LIKE "A LIST" OR "A MAP"; JUST AS A LIST CAN BE IMPLEMENTED

The operation of adding an element to the rear of the queue is known as enqueue, and the operation of removing an element from the front is known as dequeue. Other operations may also be allowed, often including a peek or front operation that returns the value of the next element to be dequeued without dequeuing

In a FIFO data structure, the first element added to the queue will be the first one to be removed. This is equivalent to the requirement that once a new element is added, all elements that were added before have to be removed before the new element can be removed. A queue is an example of a linear data structure, or more abstractly a sequential collection. Queues are common in computer programs, where they are implemented as data structures coupled with access routines, as an abstract data structure or in object-oriented languages as classes. Common implementations are circular buffers and linked lists. Queues provide services in computer science, transport, and operations research where various entities such as data, objects, persons, or events are stored and held to be processed later. In these contexts, the queue performs the function of a buffer. Another usage of queues is in the implementation of breadth-first search. Theoretically, one characteristic of a queue is that it does not have a specific capacity. Regardless of how many elements

The simple trick of turning the array into a closed circle and letting the head and tail drift around endlessly in that circle makes it unnecessary to ever move items stored in the array. If n is the size of the array, then computing indices modulo n will turn the array into a circle. This is still the conceptually simplest way to construct a queue in a high-level language, but it does admittedly slow things down a little, because the array indices must be compared to zero and the array size, which is comparable to the time taken to check whether an array index is out of bounds, which some languages do, but this will certainly be the method of choice for a quick and dirty implementation, or for any high-level language that does not have pointer syntax. The array size must be declared ahead of time, but some implementations simply double the declared array size when overflow occurs. Most modern languages with objects or pointers can implement or come with libraries for dynamic lists. Such data structures may have not specified a fixed capacity limit besides memory constraints. Queue overflow results from trying to add an element onto a full queue

IN COMPUTER SCIENCE, A QUEUE IS A COLLECTION OF ENTITIES THAT ARE MAINTAINED IN A SEQUENCE AND CAN BE MODIFIED BY THE ADDITION OF ENTITIES AT ONE END OF THE SEQUENCE AND THE REMOVAL OF ENTITIES FROM THE OTHER END OF THE SEQUENCE. BY CONVENTION, THE END OF THE SEQUENCE AT WHICH ELEMENTS ARE ADDED IS CALLED

THE OPERATION OF ADDING AN ELEMENT TO THE REAR OF THE QUEUE IS KNOWN AS ENQUEUE, AND THE OPERATION OF REMOVING AN ELEMENT FROM THE FRONT IS KNOWN AS DEQUEUE. OTHER OPERATIONS MAY ALSO BE ALLOWED, OFTEN INCLUDING A PEEK OR FRONT OPERATION THAT RETURNS THE VALUE OF THE NEXT ELEMENT TO BE DEQUEUED WITHOUT DEQUEUING IT. THE OPERATIONS OF A QUEUE MAKE IT A FIRST IN FIRST OUT (FIFO) DATA

In computer science, a queue is a collection of entities that are maintained in a sequence and can be modified by the addition of entities at one end of the sequence and the removal of entities from the other end of the sequence. By convention, the end of the sequence at which elements are added is called

The operation of adding an element to the rear of the queue is known as enqueue, and the operation of removing an element from the front is known as dequeue. Other operations may also be allowed, often including a peek or front operation that returns the value of the next element to be dequeued without dequeuing it. The operations of a queue make it a first in first out (FIFO) data

IN COMPUTER SCIENCE, A QUEUE IS A COLLECTION OF ENTITIES that are maintained in a sequence and can be modified by the addition of entities at one end of the sequence and the removal of entities from the other end of the sequence. By convention, the end of the sequence at which elements are added is called the back, tail, or

IN A PRIORITY QUEUE, AN ELEMENT WITH HIGH PRIORITY IS SERVED BEFORE AN ELEMENT WITH LOW PRIORITY. IN SOME IMPLEMENTATIONS, IF TWO ELEMENTS HAVE THE SAME PRIORITY, THEY ARE SERVED ACCORDING TO THE ORDER IN WHICH THEY WERE ENQUEUED, WHILE IN OTHER IMPLEMENTATIONS, ORDERING OF ELEMENTS WITH THE SAME PRIORITY IS UNDEFINED. WHILE PRIORITY QUEUES ARE OFTEN IMPLEMENTED WITH HEAPS, THEY ARE CONCEPTUALLY DISTINCT FROM HEAPS. A PRIORITY QUEUE IS A CONCEPT LIKE "A LIST" OR "A MAP"; JUST AS A LIST CAN BE IMPLEMENTED

The operation of adding an element to the rear of the queue is known as enqueue, and the operation of removing an element from the front is known as dequeue. Other operations may also be allowed, often including a peek or front operation that returns the value of the next element to be dequeued without dequeuing

In a FIFO data structure, the first element added to the queue will be the first one to be removed. This is equivalent to the requirement that once a new element is added, all elements that were added before have to be removed before the new element can be removed. A queue is an example of a linear data structure, or more abstractly a sequential collection. Queues are common in computer programs, where they are implemented as data structures coupled with access routines, as an abstract data structure or in object-oriented languages as classes. Common implementations are circular buffers and linked lists. Queues provide services in computer science, transport, and operations research where various entities such as data, objects, persons, or events are stored and held to be processed later. In these contexts, the queue performs the function of a buffer. Another usage of queues is in the implementation of breadth-first search. Theoretically, one characteristic of a queue is that it does not have a specific capacity. Regardless of how many elements

The simple trick of turning the array into a closed circle and letting the head and tail drift around endlessly in that circle makes it unnecessary to ever move items stored in the array. If n is the size of the array, then computing indices modulo n will turn the array into a circle. This is still the conceptually simplest way to construct a queue in a high-level language, but it does admittedly slow things down a little, because the array indices must be compared to zero and the array size, which is comparable to the time taken to check whether an array index is out of bounds, which some languages do, but this will certainly be the method of choice for a quick and dirty implementation, or for any high-level language that does not have pointer syntax. The array size must be declared ahead of time, but some implementations simply double the declared array size when overflow occurs. Most modern languages with objects or pointers can implement or come with libraries for dynamic lists. Such data structures may have not specified a fixed capacity limit besides memory constraints. Queue overflow results from trying to add an element onto a full queue

IN COMPUTER SCIENCE, A QUEUE IS A COLLECTION OF ENTITIES THAT ARE MAINTAINED IN A SEQUENCE AND CAN BE MODIFIED BY THE ADDITION OF ENTITIES AT ONE END OF THE SEQUENCE AND THE REMOVAL OF ENTITIES FROM THE OTHER END OF THE SEQUENCE. BY CONVENTION, THE END OF THE SEQUENCE AT WHICH ELEMENTS ARE ADDED IS CALLED

THE OPERATION OF ADDING AN ELEMENT TO THE REAR OF THE QUEUE IS KNOWN AS ENQUEUE, AND THE OPERATION OF REMOVING AN ELEMENT FROM THE FRONT IS KNOWN AS DEQUEUE. OTHER OPERATIONS MAY ALSO BE ALLOWED, OFTEN INCLUDING A PEEK OR FRONT OPERATION THAT RETURNS THE VALUE OF THE NEXT ELEMENT TO BE DEQUEUED WITHOUT DEQUEUING IT. THE OPERATIONS OF A QUEUE MAKE IT A FIRST IN FIRST OUT (FIFO) DATA

In computer science, a queue is a collection of entities that are maintained in a sequence and can be modified by the addition of entities at one end of the sequence and the removal of entities from the other end of the sequence. By convention, the end of the sequence at which elements are added is called

The operation of adding an element to the rear of the queue is known as enqueue, and the operation of removing an element from the front is known as dequeue. Other operations may also be allowed, often including a peek or front operation that returns the value of the next element to be dequeued without dequeuing it. The operations of a queue make it a first in first out (FIFO) data

IN COMPUTER SCIENCE, A QUEUE IS A COLLECTION OF ENTITIES that are maintained in a sequence and can be modified by the addition of entities at one end of the sequence and the removal of entities from the other end of the sequence. By convention, the end of the sequence at which elements are added is called the back, tail, or

IN A PRIORITY QUEUE, AN ELEMENT WITH HIGH PRIORITY IS SERVED BEFORE AN ELEMENT WITH LOW PRIORITY. IN SOME IMPLEMENTATIONS, IF TWO ELEMENTS HAVE THE SAME PRIORITY, THEY ARE SERVED ACCORDING TO THE ORDER IN WHICH THEY WERE ENQUEUED, WHILE IN OTHER IMPLEMENTATIONS, ORDERING OF ELEMENTS WITH THE SAME PRIORITY IS UNDEFINED. WHILE PRIORITY QUEUES ARE OFTEN IMPLEMENTED WITH HEAPS, THEY ARE CONCEPTUALLY DISTINCT FROM HEAPS. A PRIORITY QUEUE IS A CONCEPT LIKE "A LIST" OR "A MAP"; JUST AS A LIST CAN BE IMPLEMENTED

The operation of adding an element to the rear of the queue is known as enqueue, and the operation of removing an element from the front is known as dequeue. Other operations may also be allowed, often including a peek or front operation that returns the value of the next element to be dequeued without dequeuing

In a FIFO data structure, the first element added to the queue will be the first one to be removed. This is equivalent to the requirement that once a new element is added, all elements that were added before have to be removed before the new element can be removed. A queue is an example of a linear data structure, or more abstractly a sequential collection. Queues are common in computer programs, where they are implemented as data structures coupled with access routines, as an abstract data structure or in object-oriented languages as classes. Common implementations are circular buffers and linked lists. Queues provide services in computer science, transport, and operations research where various entities such as data, objects, persons, or events are stored and held to be processed later. In these contexts, the queue performs the function of a buffer. Another usage of queues is in the implementation of breadth-first search. Theoretically, one characteristic of a queue is that it does not have a specific capacity. Regardless of how many elements

The simple trick of turning the array into a closed circle and letting the head and tail drift around endlessly in that circle makes it unnecessary to ever move items stored in the array. If n is the size of the array, then computing indices modulo n will turn the array into a circle. This is still the conceptually simplest way to construct a queue in a high-level language, but it does admittedly slow things down a little, because the array indices must be compared to zero and the array size, which is comparable to the time taken to check whether an array index is out of bounds, which some languages do, but this will certainly be the method of choice for a quick and dirty implementation, or for any high-level language that does not have pointer syntax. The array size must be declared ahead of time, but some implementations simply double the declared array size when overflow occurs. Most modern languages with objects or pointers can implement or come with libraries for dynamic lists. Such data structures may have not specified a fixed capacity limit besides memory constraints. Queue overflow results from trying to add an element onto a full queue

IN COMPUTER SCIENCE, A QUEUE IS A COLLECTION OF ENTITIES THAT ARE MAINTAINED IN A SEQUENCE AND CAN BE MODIFIED BY THE ADDITION OF ENTITIES AT ONE END OF THE SEQUENCE AND THE REMOVAL OF ENTITIES FROM THE OTHER END OF THE SEQUENCE. BY CONVENTION, THE END OF THE SEQUENCE AT WHICH ELEMENTS ARE ADDED IS CALLED

THE OPERATION OF ADDING AN ELEMENT TO THE REAR OF THE QUEUE IS KNOWN AS ENQUEUE, AND THE OPERATION OF REMOVING AN ELEMENT FROM THE FRONT IS KNOWN AS DEQUEUE. OTHER OPERATIONS MAY ALSO BE ALLOWED, OFTEN INCLUDING A PEEK OR FRONT OPERATION THAT RETURNS THE VALUE OF THE NEXT ELEMENT TO BE DEQUEUED WITHOUT DEQUEUING IT. THE OPERATIONS OF A QUEUE MAKE IT A FIRST IN FIRST OUT (FIFO) DATA

In computer science, a queue is a collection of entities that are maintained in a sequence and can be modified by the addition of entities at one end of the sequence and the removal of entities from the other end of the sequence. By convention, the end of the sequence at which elements are added is called

The operation of adding an element to the rear of the queue is known as enqueue, and the operation of removing an element from the front is known as dequeue. Other operations may also be allowed, often including a peek or front operation that returns the value of the next element to be dequeued without dequeuing it. The operations of a queue make it a first in first out (FIFO) data

IN COMPUTER SCIENCE, A QUEUE IS A COLLECTION OF ENTITIES that are maintained in a sequence and can be modified by the addition of entities at one end of the sequence and the removal of entities from the other end of the sequence. By convention, the end of the sequence at which elements are added is called the back, tail, or

IN A PRIORITY QUEUE, AN ELEMENT WITH HIGH PRIORITY IS SERVED BEFORE AN ELEMENT WITH LOW PRIORITY. IN SOME IMPLEMENTATIONS, IF TWO ELEMENTS HAVE THE SAME PRIORITY, THEY ARE SERVED ACCORDING TO THE ORDER IN WHICH THEY WERE ENQUEUED, WHILE IN OTHER IMPLEMENTATIONS, ORDERING OF ELEMENTS WITH THE SAME PRIORITY IS UNDEFINED. WHILE PRIORITY QUEUES ARE OFTEN IMPLEMENTED WITH HEAPS, THEY ARE CONCEPTUALLY DISTINCT FROM HEAPS. A PRIORITY QUEUE IS A CONCEPT LIKE "A LIST" OR "A MAP"; JUST AS A LIST CAN BE IMPLEMENTED

The operation of adding an element to the rear of the queue is known as enqueue, and the operation of removing an element from the front is known as dequeue. Other operations may also be allowed, often including a peek or front operation that returns the value of the next element to be dequeued without dequeuing

In a FIFO data structure, the first element added to the queue will be the first one to be removed. This is equivalent to the requirement that once a new element is added, all elements that were added before have to be removed before the new element can be removed. A queue is an example of a linear data structure, or more abstractly a sequential collection. Queues are common in computer programs, where they are implemented as data structures coupled with access routines, as an abstract data structure or in object-oriented languages as classes. Common implementations are circular buffers and linked lists. Queues provide services in computer science, transport, and operations research where various entities such as data, objects, persons, or events are stored and held to be processed later. In these contexts, the queue performs the function of a buffer. Another usage of queues is in the implementation of breadth-first search. Theoretically, one characteristic of a queue is that it does not have a specific capacity. Regardless of how many elements

The simple trick of turning the array into a closed circle and letting the head and tail drift around endlessly in that circle makes it unnecessary to ever move items stored in the array. If n is the size of the array, then computing indices modulo n will turn the array into a circle. This is still the conceptually simplest way to construct a queue in a high-level language, but it does admittedly slow things down a little, because the array indices must be compared to zero and the array size, which is comparable to the time taken to check whether an array index is out of bounds, which some languages do, but this will certainly be the method of choice for a quick and dirty implementation, or for any high-level language that does not have pointer syntax. The array size must be declared ahead of time, but some implementations simply double the declared array size when overflow occurs. Most modern languages with objects or pointers can implement or come with libraries for dynamic lists. Such data structures may have not specified a fixed capacity limit besides memory constraints. Queue overflow results from trying to add an element onto a full queue

In computer science, a queue is a collection of entities that are maintained in a sequence and can be modified by the addition of entities at one end of the sequence and the removal of entities from the other end of the sequence. By convention, the end of the sequence at which elements are added is called the back, tail, or rear of the queue, and the end at which elements are removed is called the head or front of the queue, analogously to the words used when people line up to wait for goods or services. The operation of adding an element to the rear of the queue is known as

In computer science, a queue is a collection of entities that are maintained in a sequence and can be modified by the addition of entities at one end of the **sequence** and the removal of entities from the other end of the sequence. By convention, the end of the sequence at which **elements** are added is called the back, tail, or rear of the queue, and the end at which elements are removed is called the head or front **of the queue, analogously to the words used when people line up to wait for goods or services. The operation of adding an element to the rear of the queue is known as**

In computer science, a queue is a collection of entities that are maintained in a sequence and can be modified by the addition of entities at one end of the **sequence** and the removal of entities from the other end of the sequence. By convention, the end of the sequence at which **elements** are added is called the back, tail, or rear of the queue, and the end at which elements are removed is called the head or front **of the queue, analogously to the words used when people line up to wait for goods or services. The operation of adding an element to the rear of the queue is known as**

In computer science, a queue is a collection of entities that are maintained in a sequence and can be modified by the addition of entities at one end of the **sequence** and the removal of entities from the other end of the sequence. By convention, the end of the sequence at which **elements** are added is called the back, tail, or rear of the queue, and the end at which elements are removed is called the head or front **of the queue, analogously to the words used when people line up to wait for goods or services. The operation of adding an element to the rear of the queue is known as**

OpenType Features

DOTTED ZERO	9870123 9870123
BIG ASTERISK	9*10=90 9*10=90
UPPERCASE FORMS	@TYPESUPPLY @TYPESUPPLY
CONTEXTUAL ALTERNATES	Hyphenated-Word Hyphenated-Word

Character Set

UPPERCASE	ABCDEFGHIJKLMNOPQRSTUVWXYZ ÆŒØĐÞĦŁŊŦÀÁÂÃÄÅĀĂĄÇĆĊČĎĐÈÉÊËĒĔĖĘĚẼĞĠ ĢĦÌÍÎÏĨĪĬĮİĶĹĻĽĿŁÑŃŅŇŊÒÓÔÕÖŌŎŐŔŖŘŚŞŠ ȘŤŦÙÚÛÜŨŪŬŮŰŲŴẀẂẄÝŶỲŸỸŹŻŽǼIJ
LOWERCASE	abcdefgghijklmnopqrstuvwxyz ßæœøðđþħłŋŧàáâãäåāăąçćċčďđèéêëēĕėęěẽ ğġģğġģħìíîïĩīĭįķĺļľŀłñńņňŋòóôõöōŏőŕŗ řśşšșťŧùúûüũūŭůűųŵẁẃẅýÿŷỳỹźżžǽ
FIGURES	0123456789
DOTTED ZERO	0
FIGURE STUFF	$¢£¥₩€ƒ +−±×÷=≠≈<>≤≥ °#%‰
PUNCTUATION	.,:;…!¡?¿ -–—·•_'"“”‘’„‚«»‹› /\\|¦()[]{}
UPPERCASE	¡¿
SYMBOLS & REFERENCE MARKS	&@ªº *†‡~^ ¶§©℗®™
BIG ASTERISK	*
UPPERCASE	@ª
ARROWS	←→↑↓↖↗↘↙

Supported Languages

English

In Closing

CONTACT

Type Supply
122 Overbrook Rd.
Baltimore, MD 21212
United States
info@typesupply.com
typesupply.com

LEGAL STUFF

TEXT

wikipedia.org